```
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
UUU          UUU   AAA         AAA   FFF
UUU          UUU   AAA         AAA   FFF
UUU          UUU   AAA         AAA   FFF
UUUUUUUUUUUUUUU    AAA         AAA   FFF
UUUUUUUUUUUUUUU    AAA         AAA   FFF
UUUUUUUUUUUUUUU    AAA         AAA   FFF
```

```
UU      UU    AAAAAA    FFFFFFFFFF  MM      MM    AAAAAA     IIIIII    NN      NN
UU      UU    AAAAAA    FFFFFFFFFF  MM      MM    AAAAAA     IIIIII    NN      NN
UU      UU   AA    AA   FF          MMMM  MMMM   AA    AA      II      NN      NN
UU      UU   AA    AA   FF          MMMM  MMMM   AA    AA      II      NN      NN
UU      UU   AA    AA   FF          MM  MM  MM   AA    AA      II      NNNN    NN
UU      UU   AA    AA   FF          MM  MM  MM   AA    AA      II      NNNN    NN
UU      UU   AA    AA   FFFFFFFF    MM      MM   AA    AA      II      NN  NN  NN
UU      UU   AA    AA   FFFFFFFF    MM      MM   AA    AA      II      NN  NN  NN
UU      UU   AAAAAAAAAA FF          MM      MM   AAAAAAAAAA      II      NN    NNNN
UU      UU   AAAAAAAAAA FF          MM      MM   AAAAAAAAAA      II      NN    NNNN
UU      UU   AA    AA   FF          MM      MM   AA    AA      II      NN      NN
UU      UU   AA    AA   FF          MM      MM   AA    AA      II      NN      NN
UUUUUUUUUU   AA    AA   FF          MM      MM   AA    AA    IIIIII    NN      NN
UUUUUUUUUU   AA    AA   FF          MM      MM   AA    AA    IIIIII    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 uafmain (main = start,
;     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 user
;    38        0038  1 !       authorization file which contains usernames, passwords, quotas,
;    39        0039  1 !       and defaults.  The following functions are provided:
;    40        0040  1 !
;    41        0041  1 !               ADD - add a new user record to the authorization file (UAF)
;    42        0042  1 !               COPY - copy a user record, give copied record a new name
;    43        0043  1 !               DEFAULT - change a default value
;    44        0044  1 !               EXIT - exit program and update file
;    45        0045  1 !               HELP - ask for explanation
;    46        0046  1 !               LIST - complete list of records to file
;    47        0047  1 !               MODIFY - change one or more values for a user
;    48        0048  1 !               REMOVE - remove a user record
;    49        0049  1 !               RENAME - rename a user record (COPY; REMOVE)
;    50        0050  1 !               SHOW - display the values from a user record
;    51        0051  1 !
;    52        0052  1 ! ENVIRONMENT:
;    53        0053  1 !
;    54        0054  1 ! AUTHOR:       Henry M. Levy, CREATION DATE: 1-June-1977
;    55        0055  1 !
;    56        0056  1 ! MODIFIED BY:
;    57        0057  1 !
```

```
;    58     0058  1 ;        V03-024 JRL0036          John R. Lawson, Jr.     07-Aug-1984 17:08
;    59     0059  1 ;                Hide UPGRADE, DOWNGRADE, TMPJNL, PRMJNL privileges.  This
;    60     0060  1 ;                is a temporary work-around; it is marked in the right-hand
;    61     0061  1 ;                margin with !** in routine PRINT_PRIV.
;    62     0062  1 ;
;    63     0063  1 ;        V03-023 JRL0027          John R. Lawson, Jr.     25-Jul-1984 11:01
;    64     0064  1 ;                Add MODIFY/SYSTEM_PASSWORD=xxxxx to modify the system
;    65     0065  1 ;                password.
;    66     0066  1 ;
;    67     0067  1 ;        V03-022 JRL0029          John R. Lawson, Jr.     25-Jul-1984 10:47
;    68     0068  1 ;                Find better names for UAF$_NEWMSG10 and UAF$_NEWMSG15
;    69     0069  1 ;
;    70     0070  1 ;        V03-021 JRL0020          John R. Lawson, Jr.     09-Jul-1984 15:51
;    71     0071  1 ;                Bark when a proxy name is changed with the RENAME command.
;    72     0072  1 ;
;    73     0073  1 ;        V03-020 JRL0017          John R. Lawson, Jr.     02-Jul-1984 22:13
;    74     0074  1 ;                Modify GET_UAF_RECORD so that it does not get the system
;    75     0075  1 ;                password record.
;    76     0076  1 ;
;    77     0077  1 ;        V03-029 JRL0013          John R. Lawson, Jr.     02-Jul-1984 12:28
;    78     0078  1 ;                Display privileges of users in groups less than .EXE$GL_SYSUIC
;    79     0079  1 ;                as "ALL".
;    80     0080  1 ;
;    81     0081  1 ;        V03-028 JRL0010          John R. Lawson, Jr.     25-Jun-1984 15:56
;    82     0082  1 ;                Changed 'X'/'-' to '#'/'-' in primary/secondary access
;    83     0083  1 ;                display.
;    84     0084  1 ;
;    85     0085  1 ;        V03-027 JRL0008          John R. Lawson, Jr.     21-Jun-1984 14:00
;    86     0086  1 ;                Add support for the /PWDEXPIRED qualifier (pre-expired
;    87     0087  1 ;                password).
;    88     0088  1 ;
;    89     0089  1 ;        V03-026 JRL0006          John R. Lawson, Jr.     20-Jun-1984 12:28
;    90     0090  1 ;                Obliterate operator's console messages
;    91     0091  1 ;
;    92     0092  1 ;        V03-025 JRL0002          John R. Lawson, Jr.     15-Jun-1984 09:55
;    93     0093  1 ;                Change all internal messages to calls to LIB$SIGNAL and
;    94     0094  1 ;                the message utility -- place all messages in UAFMSG.MSG
;    95     0095  1 ;
;    96     0096  1 ;        V03-024 LY0494           Larry Yetto             11-JUN-1984 12:59
;    97     0097  1 ;                Fix noise error message comming from ADD/ID/USER=* caused
;    98     0098  1 ;                by a flag not properly being set
;    99     0099  1 ;
;   100     0100  1 ;        V03-023 MHB0150          Mark Bramhall           2-May-1984
;   101     0101  1 ;                Remove unused reference to TPARSE definitions.
;   102     0102  1 ;                Add DISRECONNECT flag.
;   103     0103  1 ;                Add security auditing for SYSUAF/NETUAF changes.
;   104     0104  1 ;
;   105     0105  1 ;        V03-022 LY0474           Larry Yetto             9-APR-1984 08:32
;   106     0106  1 ;                Zero the login failure and last login fields
;   107     0107  1 ;                on a copy operation.
;   108     0108  1 ;
;   109     0109  1 ;        V03-021 LY0466           Larry Yetto             22-MAR-1984 13:52
;   110     0110  1 ;                Add support for the rights data base functions
;   111     0111  1 ;
;   112     0112  1 ;        V03-020 ACG0397          Andrew C. Goldstein,    24-Feb-1984  23:21
;   113     0113  1 ;                Clean up display formatting
;   114     0114  1 ;
```

```
;  115          0115  1 ;
;  116          0116  1 ;       V03-019 ACG0397         Andrew C. Goldstein,    6-Feb-1984  16:27
;  117          0117  1 ;               Add DISREPORT to flags, clean up record locking
;  118          0118  1 ;
;  119          0119  1 ;       V03-018 ACG0388         Andrew C. Goldstein,    12-Jan-1984  19:21
;  120          0120  1 ;               Add command input to handle new UAF features;
;  121          0121  1 ;               general code cleanup
;  122          0122  1 ;
;  123          0123  1 ;       V03-017 ACG0385         Andrew C. Goldstein,    6-Jan-1984  18:28
;  124          0124  1 ;               V4 UAF format change; remove read-only under installed
;  125          0125  1 ;               SYSPRV feature; misc. code cleanups
;  126          0126  1 ;
;  127          0127  1 ;       V03-016 TMK0001         Todd M. Katz            10-Oct-1983
;  128          0128  1 ;               Add JTQUOTA (job-wide logical name table creation quota)
;  129          0129  1 ;               qualifier.
;  130          0130  1 ;
;  131          0131  1 ;       V03-015 LMP0153         L. Mark Pilant,         13-Sep-1983  11:57
;  132          0132  1 ;               Add minimal support for alphanumeric UICs.
;  133          0133  1 ;
;  134          0134  1 ;       014     JWT0105         Jim Teague      30-Mar-1983
;  135          0135  1 ;               Small changes to CLITABLES implementation.
;  136          0136  1 ;
;  137          0137  1 ;       013     JWT0104         Jim Teague      29-Mar-1983
;  138          0138  1 ;               Add CLITABLES qualifier.
;  139          0139  1 ;
;  140          0140  1 ;       012     WMC0001         Wayne Cardoza   15-Mar-1983
;  141          0141  1 ;               Add MAXDETACH qualifier.
;  142          0142  1 ;
;  143          0143  1 ;       011     JWT0097         Jim Teague      23-Feb-1983
;  144          0144  1 ;               Fix RENAME problem with proxy entries.
;  145          0145  1 ;
;  146          0146  1 ;       010     JWT0096         Jim Teague      08-Feb-1983
;  147          0147  1 ;               Log NETUAF changes to console, too.
;  148          0148  1 ;
;  149          0149  1 ;       009     JWT0087         Jim Teague      11-Jan-1983
;  150          0150  1 ;               Change SYSWSQUOTA for created UAFs to 350
;  151          0151  1 ;
;  152          0152  1 ;       008     JWT0082         Jim Teague      05-Jan-1983
;  153          0153  1 ;               Fix problem with LIST/PROXY.
;  154          0154  1 ;
;  155          0155  1 ;       007     JWT0079         Jim Teague      15-Dec-1982
;  156          0156  1 ;               Enlarge output field for BYTLM; reset pending
;  157          0157  1 ;               mail count for COPY operations.
;  158          0158  1 ;
;  159          0159  1 ;       006     JWT0072         Jim Teague      03-Dec-1982
;  160          0160  1 ;               Add global longword which can be patched to
;  161          0161  1 ;               enable/disable console logging of SYSUAF mods.
;  162          0162  1 ;
;  163          0163  1 ;       005     JWT0069         Jim Teague      24-Nov-1982
;  164          0164  1 ;               Allow redefinition of sys$output.
;  165          0165  1 ;
;  166          0166  1 ;       004     JWT0057         Jim Teague      21-Sep-1982
;  167          0167  1 ;               Add a message to tell whether or not NETUAF was
;  168          0168  1 ;               modified.
;  169          0169  1 ;
;  170          0170  1 ;       003     JWT0042         Jim Teague      15-Jul-1982
;  171          0171  1 ;               Make SYSUAF.LIS and NETUAF.LIS world noread.
```

```
;  172          0172  1 !
;  173          0173  1 !       002     JWT0036         Jim Teague      08-Jun-1982
;  174          0174  1 !               Add full wildcarding to show/proxy
;  175          0175  1 !
;  176          0176  1 !       001     JWT0022         Jim Teague      17-Mar-1982
;  177          0177  1 !               Fix bug that caused failure to reparse command line for
;  178          0178  1 !               wildcard modifications.  List default device on its own
;  179          0179  1 !               line for show/full.
;  180          0180  1 !--
```

```
;  182     0181  1 !
;  183     0182  1 ! Require files
;  184     0183  1 !
;  185     0184  1 require
;  186     0185  1         'lib$:uafreq';
;  187     0281  1
;  188     0282  1 !
;  189     0283  1 ! INCLUDE FILES:
;  190     0284  1 !
;  191     0285  1 library 'SYS$LIBRARY:LIB.L32';
;  192     0286  1
;  193     0287  1 !
;  194     0288  1 ! TABLE OF CONTENTS:
;  195     0289  1 !
;  196     0290  1
;  197     0291  1 forward routine
;  198     0292  1         start,                          ! controlling code
;  199     0293  1         setup           : novalue,      ! open initial files
;  200     0294  1         add_uaf         : novalue,      ! insert new user record
;  201     0295  1         add_proxy       : novalue,      ! insert new proxy record
;  202     0296  1         remote_parse,                   ! parses "node: : remoteuser"
;  203     0297  1         copy_uaf,                       ! copy user record
;  204     0298  1         create_proxy    : novalue,      ! create NETUAF.DAT proxy file
;  205     0299  1         modify_uaf      : novalue,      ! update user record(s)
;  206     0300  1         modify_rec,                     ! update a user record action routine
;  207     0301  1         remove_uaf      : novalue,      ! remove username from file
;  208     0302  1         remove_proxy    : novalue,      ! remove a proxy record
;  209     0303  1         rename_uaf      : novalue,      ! rename user record
;  210     0304  1         adjust_proxy    : novalue,      ! implicitly remove/update proxy record
;  211     0305  1         default_uaf     : novalue,      ! change default record
;  212     0306  1         list_proxy      : novalue,      ! list proxy entries in NETUAF.LIS
;  213     0307  1         list_uaf        : novalue,      ! list file routine
;  214     0308  1         show_user_uaf   : novalue,      ! display user record
;  215     0309  1         show_proxy      : novalue,      ! display proxy record at terminal
;  216     0310  1         locate_proxy,                   ! access given proxy record(s)
;  217     0311  1         get_proxy_record,               ! read single proxy record
;  218     0312  1         display_proxy   : novalue,      ! format and output a proxy entry
;  219     0313  1         wild_user,                      ! user wild card routine
;  220     0314  1         display_brief,                  ! writes a brief user display
;  221     0315  1         classify_priv,                  ! classifies contents of priv vector
;  222     0316  1         display_full,                   ! writes the full user display
;  223     0317  1         display_hours   : novalue,      ! display hourly restrictions
;  224     0318  1         convert_time    : novalue,      ! convert time value to string
;  225     0319  1         print_priv      : novalue,      ! print privilege bits
;  226     0320  1         build_ini_recs  : novalue,      ! build initial file records for default
;  227     0321  1                                         ! and system manager
;  228     0322  1         get_user_record,                ! get username and lookup record
;  229     0323  1         locate_user,                    ! lookup user record in UAF
;  230     0324  1         get_uaf_record,                 ! routine to deal with record locking
;  231     0325  1         get_cmd_line,                   ! input user command line
;  232     0326  1         ask             : novalue,      ! prompt terminal for input
;  233     0327  1         fmt_sys_msg     : novalue,      ! output system message file message
;  234     0328  1         faoout,                         ! output formatted message
;  235     0329  1         help_uaf        : novalue,      ! help routine
;  236     0330  1         exit_uaf        : novalue,      ! normal exit routine
;  237     0331  1         SIGNAL_SYNTAX   : novalue,      ! missing qualifier
;  238     0332  1         acc$exit        : novalue,      ! exit and cleanup routine
```

```
;  239     0333  1          uaf$mod_sys_pwd : novalue,      ! modify the system password
;  240     0334  1          security_audit  : novalue;      ! Perform a security audit
;  241     0335  1
;  242     0336  1  linkage
;  243     0337  1          fmg_match = jsb (register = 2, register = 3, register = 4,
;  244     0338  1                           register = 5) : notused (10, 11);
;  245     0339  1
;  246     0340  1  !
;  247     0341  1  ! EXTERNAL REFERENCES:
;  248     0342  1  !
;  249     0343  1
;  250     0344  1  external literal
;  251     0345  1          cli$_bufovf,
;  252     0346  1          cli$_noclint;
;  253     0347  1
;  254     0348  1
;  255     0349  1  external routine
;  256     0350  1          lbr$output_help,
;  257     0351  1          lib$get_foreign,
;  258     0352  1          lib$get_input,
;  259     0353  1          lib$put_output,
;  260     0354  1          fmg$match_name  : fmg_match,
;  261     0355  1          cli$dcl_parse,
;  262     0356  1          cli$dispatch,
;  263     0357  1          cli$present,
;  264     0358  1          cli$get_value,
;  265     0359  1          update_record,                 ! modify all specified fields
;  266     0360  1          parse_wild,                    ! parses a wildcarded user specification
;  267     0361  1          lgi$hpwd,                      ! hash password routine
;  268     0362  1          uaf$add_ident_recbuf,
;  269     0363  1          uaf$build_holder,
;  270     0364  1          uaf$find_uic ,
;  271     0365  1          uaf$remove_ident_recbuf,
;  272     0366  1          uaf$write_rights;
;  273     0367  1
;  274     0368  1  external
;  275     0369  1          EXE$GL_SYSUIC   : long ,
;  276     0370  1          rdb_header_flag : byte ,
;  277     0371  1          rdb_list_flag   : byte ,
;  278     0372  1          attributes      : long ,
;  279     0373  1          holder          : $bblock[8] ,
;  280     0374  1          ident           : $bblock [4] ,
;  281     0375  1          authorize_commands,            ! AUTHORIZE command parse tables
;  282     0376  1          prv$ab_names;                  ! address of privilege name table
;  283     0377  1
;  284     0378  1
;  285     0379  1  !
;  286     0380  1  ! MACROS:
;  287     0381  1  !
;  288     0382  1  macro
;  289     0383  1
;  290   M 0384  1      namelen (x, y) =
;  291   M 0385  1          begin
;  292   M 0386  1          builtin
;  293   M 0387  1              locc;
;  294   M 0388  1          register
;  295   M 0389  1              r0 = 0;
```

```
;  296     M 0390  1          locc (%ref(' '), %ref(x), y; r0);
;  297     M 0391  1          x - .r0
;  298       0392  1          end%,
;  299       0393  1
;  300     M 0394  1      cstring[] = (uplit byte (%charcount (%string (%remaining)),
;  301       0395  1                   %string (%remaining)))%,
;  302       0396  1
;  303       0397  1      fatal[] = (fmt_sys_msg (%remaining); acc$exit ())%,
;  304       0398  1  !
;  305       0399  1  ! Macros to check for success or failure from RMS
;  306       0400  1  !
;  307       0401  1      rmsbad (string) = (not (rmserr = string)) %,
;  308       0402  1      rmsok (string) = (rmserr = string) %,
;  309       0403  1  !
;  310       0404  1  ! Macros to set up and write an FAO string.
;  311       0405  1  !
;  312     M 0406  1      faomac (faomsg)[] =
;  313     M 0407  1          begin
;  314     M 0408  1          faodsc[dsc$w_length] = . (faomsg)<0,8>;
;  315     M 0409  1          faodsc[dsc$a_pointer] = (faomsg) + 1;
;  316     M 0410  1          $fao (faodsc, rabptr[rab$w_rsz], disdsc $comma (%remaining)
;  317     M 0411  1                       %remaining);
;  318     M 0412  1          $put (rab = .rabptr);
;  319       0413  1          end %,
;  320       0414  1
;  321       0415  1      $comma[] =, %,
;  322       0416  1
;  323     M 0417  1      output_null =
;  324     M 0418  1          begin
;  325     M 0419  1          rabptr[rab$w_rsz] = 0;
;  326     M 0420  1          $put (rab = .rabptr);
;  327       0421  1          end %,
;  328       0422  1
;  329       0423  1  !
;  330       0424  1  ! Macro to create string descriptor for command parameters and
;  331       0425  1  !       qualifiers
;  332       0426  1  !
;  333     M 0427  1      sd[a] =
;  334       0428  1          bind %name ('SD_',a) = $descriptor (a)%;
;  335       0429  1
;  336     P 0430  1       sd (
;  337     P 0431  1          'token1',       'token2',       'brief',        'full',
;  338     P 0432  1          'add_identifier',               'remove_identifier',
;  339     P 0433  1          'modify_identifier'
;  340       0434  1  );
;  341       0435  1
;  342       0436  1  field
;  343       0437  1      descr_fields =                      ! Define the fields for a DESCRIPTOR
;  344       0438  1          set
;  345       0439  1          length  = [dsc$w_length],
;  346       0440  1          dtype   = [dsc$b_dtype],
;  347       0441  1          class   = [dsc$b_class],
;  348       0442  1          pointer = [dsc$a_pointer]
;  349       0443  1          tes;
;  350       0444  1
;  351       0445  1  macro
;  352     M 0446  1      statdesc =
```

```
;  353     M 0447  1          $bblock [dsc$k_s_bln] field (descr_fields)
;  354     M 0448  1                  preset ( [length]  = 0,
;  355     M 0449  1                           [dtype]   = dsc$k_dtype_t,
;  356     M 0450  1                           [class]   = dsc$k_class_s,
;  357       0451  1                           [pointer] = 0)%;
;  358       0452  1
;  359       0453  1  macro
;  360     M 0454  1      qualstr_desc (string) =
;  361     M 0455  1          $bblock [dsc$k_s_bln] field (descr_fields)
;  362     M 0456  1                  preset ( [length]  = (%charcount(string)),
;  363     M 0457  1                           [dtype]   = dsc$k_dtype_t,
;  364     M 0458  1                           [class]   = dsc$k_class_s,
;  365       0459  1                           [pointer] = (uplit byte (%string(string))))%;
;  366       0460  1  own
;  367       0461  1      sd_attribresource   : qualstr_desc ('attributes.resource') ;
;  368       0462  1
;  369       0463  1  !
;  370       0464  1  ! EQUATED SYMBOLS:
;  371       0465  1  !
;  372       0466  1  literal
;  373       0467  1          cmdbufmax       = 508,          ! maximum command length
;  374       0468  1          false           = 0,            ! logical false
;  375       0469  1          true            = 1,            ! logical true
;  376       0470  1          update_records  = 0,            ! flag for proxy file adjustment
;  377       0471  1          remove_records  = 1,            !  "    "     "    "
;  378       0472  1          copy_flag       = 1,            ! used in routine copy_uaf
;  379       0473  1          rename_flag     = 1,            ! used in routine rename_uaf
;  380       0474  1          byte_length     = 8,            ! bits per byte
;  381       0475  1          word_length     = 16,           ! bits per word
;  382       0476  1          long_length     = 32,           ! bits per longword
;  383       0477  1          retry_rlk       = 8,            ! number of retries for a locked record
;  384       0478  1          sleep_rlk       = 500,          ! ms to sleep before retrying
;  385       0479  1          cr              = 13,
;  386       0480  1          lf              = 10,
;  387       0481  1          zero            = 0,
;  388       0482  1          cmdbuflen       = 1024;         ! size of user command buffer
;  389       0483  1
;  390       0484  1  global literal
;  391       0485  1          encrypt         = uaf$c_purdy_v, ! encryption algorithm to use
;  392       0486  1          disbuflen       = 132;          ! size of display file output buffer
;  393       0487  1
;  394       0488  1  bind
;  395       0489  1          sysuaf_string   = uplit byte ('SYSUAF'),
;  396       0490  1          netuaf_string   = uplit byte ('NETUAF'),
;  397       0491  1          mod_act_dsc     = $descriptor ('modified'),
;  398       0492  1          add_act_dsc     = $descriptor ('added'),
;  399       0493  1          rem_act_dsc     = $descriptor ('removed'),
;  400       0494  1          fao_lin_dsc     = $descriptor ('PID=!XL !AS !AS !AS record !AS on !%D'),
;  401       0495  1          dbl_colon       = $descriptor ('::'),
;  402       0496  1  !
;  403       0497  1  ! Define the system delta time to sleep before retrying to GET a locked record.
;  404       0498  1  !
;  405       0499  1          wakedelta       = uplit long (-10*1000*sleep_rlk,-1),
;  406       0500  1  !
;  407       0501  1  ! Default values for authorization file record.  These values are
;  408       0502  1  ! only used when a new authorization file is created.  If the file
;  409       0503  1  ! already exists, the default values are read from the first file
```

```
;  410     0504  1 ! record.
;  411     0505  1 !
;  412     0506  1         defuser         = cstring ('DEFAULT'),  ! default username
;  413     0507  1         defpass         = cstring ('USER'),     ! default password
;  414     0508  1         defclitabl      = cstring ('DCLTABLES'), ! default CLI tables
;  415     0509  1         defact          = cstring (''),         ! default account name
;  416     0510  1         defcli          = cstring ('DCL'),      ! default command interpreter
;  417     0511  1         defowner        = cstring (''),         ! owner's name
;  418     0512  1         deflgicmd       = cstring (''),         ! default login command file
;  419     0513  1         defgrp          = %o'200',              ! default group
;  420     0514  1         defmem          = %o'200',              ! default member
;  421     0515  1         defdir          = cstring ('[USER]'),   ! default directory name
;  422     0516  1         defdev          = cstring (''),         ! default device name
;  423     0517  1         defbiolm        = 6,                    ! default buffered I/O limit
;  424     0518  1         defbytlm        = 4096,                 ! default buffered I/O buffer space
;  425     0519  1         defdiolm        = 6,                    ! default direct I/O limit
;  426     0520  1         deffillm        = 20,                   ! default open file limit
;  427     0521  1         defflags        = 0,                    ! default flag bits
;  428     0522  1         deftqcnt        = 10,                   ! default time queue entries
;  429     0523  1         defprccnt       = 2,                    ! default subprocess count
;  430     0524  1         defpri          = 4,                    ! default process priority
;  431     0525  1         defquepri       = 4,                    ! default queue priority
;  432     0526  1         defwsquota      = 200,                  ! default working set limit
;  433     0527  1         defdfwscnt      = 150,                  ! "default" working set default size
;  434     0528  1         defwsextent     = 500,                  ! default working set extent
;  435     0529  1         defcputim       = 0,                    ! default CPU time quota
;  436     0530  1         defastlm        = 10,                   ! default AST queue limit
;  437     0531  1         defpgflquota    = 10000,                ! default paging file limit in pages
;  438     0532  1         defenqlm        = 10,                   ! default enqueue limit
;  439     0533  1         defpbytlm       = 0,                    ! default paged buffer I/O limit
;  440     0534  1         defshrfillm     = 0,                    ! default shared file limit
;  441     0535  1         defmaxjobs      = 0,                    ! default maximum concurrent jobs
;  442     0536  1         defmaxacctjobs  = 0,                    ! default maximum concurrent group jobs
;  443     0537  1         defmaxdetach    = 0,                    ! default maximum detached processes
;  444     0538  1         defjtquota      = 1024,                 ! default job-wide logical table quota
;  445     0539  2         defprimedays    = (                     ! Sat, Sun are default non-prime days
;  446     0540  2                           (1 ^ $bitposition (uaf$v_saturday)) or
;  447     0541  3                           (1 ^ $bitposition (uaf$v_sunday))
;  448     0542  1                           ),
;  449     0543  1         defhours        = 0,                    ! default all hours to allow access
;  450     0544  1                 defpriv = uplit (               ! default privilege vector
;  451     0545  2                           (
;  452     0546  2                           (1 ^ $bitposition (prv$v_netmbx)) or
;  453     0547  3                           (1 ^ $bitposition (prv$v_tmpmbx))
;  454     0548  1                           ), 0),
;  455     0549  1         defpwdlength    = 6,                    ! default min password length
;  456     0550  1         defpwdlife      = uplit (0,0),          ! default password lifetime
;  457     0551  1
;  458     0552  1 !
;  459     0553  1 ! The following are the default values for the SYSTEM user.  When
;  460     0554  1 ! a new file is created, a system manager record is created.
;  461     0555  1 !
;  462     0556  1         sysuser         = cstring ('SYSTEM'),
;  463     0557  1         syspass         = cstring ('MANAGER'),
;  464     0558  1         sysclitabl      = cstring ('DCLTABLES'),
;  465     0559  1         sysact          = cstring ('SYSTEM'),
;  466     0560  1         syscli          = cstring ('DCL'),
```

```
;  467     0561  1          sysowner        = cstring ('SYSTEM MANAGER'),
;  468     0562  1          syslgicmd       = cstring (''),
;  469     0563  1          sysgrp          = %o'1',
;  470     0564  1          sysmem          = %o'4',
;  471     0565  1          sysdir          = cstring ('[SYSMGR]'),
;  472     0566  1          sysdev          = cstring (''),
;  473     0567  1          sysbiolm        = 12,
;  474     0568  1          sysbytlm        = 20480,
;  475     0569  1          sysdiolm        = 12,
;  476     0570  1          sysfillm        = 20,
;  477     0571  1          sysflags        = 0,
;  478     0572  1          systqcnt        = 20,
;  479     0573  1          sysprccnt       = 10,
;  480     0574  1          syspri          = 4,
;  481     0575  1          sysquepri       = 4,
;  482     0576  1          syswsquota      = 350,
;  483     0577  1          syswsextent     = 1024,
;  484     0578  1          sysdfwscnt      = 150,
;  485     0579  1          syscputim       = 0,
;  486     0580  1          sysastlm        = 20,
;  487     0581  1          syspgflquota    = 10000,
;  488     0582  1          sysenqlm        = 20,
;  489     0583  1          syspbytlm       = 0,
;  490     0584  1          sysshrfillm     = 0,
;  491     0585  1          sysmaxjobs      = 0,
;  492     0586  1          sysmaxdetach    = 0,
;  493     0587  1          sysjtquota      = 1024,
;  494     0588  1          sysmaxacctjobs  = 0,
;  495     0589  2          sysprimedays    = (                     ! Sat, Sun are default non-prime days
;  496     0590  2                            (1 ^ $bitposition (uaf$v_saturday)) or
;  497     0591  3                            (1 ^ $bitposition (uaf$v_sunday))
;  498     0592  1                            ),
;  499     0593  1          syshours        = 0,
;  500     0594  1          syspriv         = uplit (rep 2 of (%x'FFFFFFFF')),
;  501     0595  1          syspwdlength    = 8,                    ! default min password length
;  502     0596  1          syspwdlife      = uplit (0,0);          ! default password lifetime
;  503     0597  1
;  504     0598  1 !
;  505     0599  1 ! GLOBAL STORAGE - must be before OWN for initialization purposes
;  506     0600  1 !
;  507     0601  1 global
;  508     0602  1          disbuf          : vector [disbuflen, byte], ! display buffer
;  509     0603  1          disdsc          : block [8, byte] initial (disbuflen, disbu[) ;
;  510     0604  1
;  511     0605  1 !
;  512     0606  1 ! OWN STORAGE:
;  513     0607  1 !
;  514     0608  1 own
;  515     0609  1          username_buf    : block [uaf$s_username, byte],
;  516     0610  1          pcb_sts         : bitvector [32],
;  517     0611  1          pid,
;  518     0612  1          username_dsc    : vector [2] initial (0, username_buf),
;  519     0613  1          recname_dsc     : vector [2],
;  520     0614  1          file_dsc        : vector [2] initial (6),
;  521     0615  1          mod_default,                            ! DEFAULT record being modified by MODIFY command
;  522     0616  1          modify_flag     : long,                 ! SYSUAF modified
;  523     0617  1          netuaf_modified,                        ! NETUAF modified
```

```
;  524          0618  1        rename_ph2      : byte initial (false),
;  525          0619  1        olduserlen      : long,
;  526          0620  1        oldusername     : vector  [uaf$s_username,byte],
;  527          0621  1        newuserlen      : long,
;  528          0622  1        newusername     : vector  [uaf$s_username,byte],
;  529          0623  1        cmdbuf          : vector [cmdbuflen, byte], ! command buffer
;  530          0624  1        default_size    : word,
;  531          0625  1        default_record  : block [uaf$c_length, byte], ! default record held here
;  532          0626  1        pwddsc          : block [8, byte],          ! password descriptor
;  533          0627  1        insize          : long,                     ! number of input chars
;  534          0628  1        brief_flag      : long,                     ! display option
;  535          0629  1        full_flag       : long,                     ! display option
;  536          0630  1        header_flag     : long,                     ! output header or not?
;  537          0631  1
;  538          0632  1
;  539        P 0633  1        infab   : $fab (                    ! FAB for terminal I/O
;  540        P 0634  1                         fac = (get, put),
;  541        P 0635  1                         rat = cr,
;  542        P 0636  1                         fnm = 'SYS$INPUT'
;  543          0637  1                         ),
;  544          0638  1
;  545        P 0639  1        inrab   : $rab (                    ! RAB for terminal I/O
;  546        P 0640  1                         rac = seq,
;  547        P 0641  1                         rop = pmt,
;  548        P 0642  1                         fab = infab
;  549          0643  1                         ),
;  550          0644  1
;  551        P 0645  1        outfab  : $fab (
;  552        P 0646  1                         fac = (put),
;  553        P 0647  1                         rat = cr,
;  554        P 0648  1                         fnm = 'SYS$OUTPUT'
;  555          0649  1                         ),
;  556          0650  1
;  557          0651  1        lstnam  : $nam (),                  ! needed for the DLT option
;  558          0652  1
;  559        P 0653  1        lstpro  : $xabpro (
;  560        P 0654  1                         pro = (rwd,rwd,rw)
;  561          0655  1                         ),
;  562          0656  1
;  563        P 0657  1        lstfab  : $fab (                    ! FAB for UAF listing file
;  564        P 0658  1                         fac = put,
;  565        P 0659  1                         rat = cr,
;  566        P 0660  1                         fnm = 'SYSUAF.LIS',
;  567        P 0661  1                         shr = nil,
;  568        P 0662  1                         org = seq,
;  569        P 0663  1                         rfm = var,
;  570        P 0664  1                         mrs = disbuflen,
;  571        P 0665  1                         nam = lstnam,
;  572        P 0666  1                         xab = lstpro
;  573          0667  1                         ),
;  574          0668  1
;  575        P 0669  1        lstrab  : $rab (                    ! RAB for UAF listing file
;  576        P 0670  1                         rac = seq,
;  577        P 0671  1                         rbf = disbuf,
;  578        P 0672  1                         fab = lstfab
;  579          0673  1                         ),
;  580          0674  1
```

```
;  581          0675  1          nlstnam : $nam (),
;  582          0676  1
;  583        P 0677  1          nlstpro : $xabpro (
;  584        P 0678  1                            pro = (rwd,rwd,rw)
;  585          0679  1                            ),
;  586          0680  1
;  587        P 0681  1          nlstfab : $fab (                    ! FAB for NETUAF listing file
;  588        P 0682  1                            fac = put,
;  589        P 0683  1                            rat = cr,
;  590        P 0684  1                            fnm = 'NETUAF.LIS',
;  591        P 0685  1                            shr = nil,
;  592        P 0686  1                            org = seq,
;  593        P 0687  1                            rfm = var,
;  594        P 0688  1                            mrs = disbuflen,
;  595        P 0689  1                            nam = nlstnam,
;  596        P 0690  1                            xab = nlstpro
;  597          0691  1                            ),
;  598          0692  1
;  599        P 0693  1          nlstrab : $rab (                    ! RAB for NETUAF listing file
;  600        P 0694  1                            rac = seq,
;  601        P 0695  1                            rbf = disbuf,
;  602        P 0696  1                            fab = nlstfab
;  603          0697  1                            ),
;  604          0698  1
;  605        P 0699  1          uafkey2 : $xabkey (                 ! XAB for User number key
;  606        P 0700  1                            kref = 2,         ! alternate key
;  607        P 0701  1                            pos0 = $byteoffset (uaf$w_mem),
;  608        P 0702  1                            siz0 = 2,
;  609        P 0703  1                            dtp = bn2,
;  610        P 0704  1                            flg = (chg,dup)
;  611          0705  1                            ),
;  612          0706  1
;  613        P 0707  1          uafkey1 : $xabkey (                 ! XAB for Group number key
;  614        P 0708  1                            kref = 1,         ! alternate key
;  615        P 0709  1                            pos0 = $byteoffset (uaf$l_uic),
;  616        P 0710  1                            siz0 = 4,
;  617        P 0711  1                            dtp = bn4,
;  618        P 0712  1                            flg = (chg,dup),
;  619        P 0713  1                            nxt = uafkey2
;  620          0714  1                            ),
;  621          0715  1
;  622        P 0716  1          uafkey0 : $xabkey (                 ! XAB for USERNAME key
;  623        P 0717  1                            kref = 0,         ! primary key
;  624        P 0718  1                            pos0 = $byteoffset (uaf$t_username),
;  625        P 0719  1                            siz0 = uaf$s_username,
;  626        P 0720  1                            nxt = uafkeyT
;  627          0721  1                            ),
;  628          0722  1
;  629        P 0723  1          uafpro  : $xabpro (                 ! XAB for file protection
;  630        P 0724  1                            pro = (rwed, rwed), ! deny world access
;  631        P 0725  1                            nxt = uafkey0
;  632          0726  1                            ),
;  633          0727  1
;  634        P 0728  1          uaffab  : $fab (                    ! FAB for work file
;  635        P 0729  1                            fop = cif,
;  636        P 0730  1                            fac = (get, put, del, upd),
;  637        P 0731  1                            fnm = 'SYSUAF',
```

```
;  638          P 0732  1                       dnm = '.DAT',
;  639          P 0733  1                       shr = (get, put, del, upd),
;  640          P 0734  1                       org = idx,
;  641          P 0735  1                       rfm = var,
;  642          P 0736  1                       mrs = uaf$c_length,
;  643          P 0737  1                       alq = 10,
;  644          P 0738  1                       deq = 10,
;  645          P 0739  1                       xab = uafpro
;  646            0740  1                       ),
;  647            0741  1 !
;  648            0742  1 ! FAB for NETUAF Proxy Login File
;  649            0743  1 !
;  650            0744  1 !
;  651          P 0745  1       nafkey1 : $xabkey (
;  652          P 0746  1                       kref = 1,
;  653          P 0747  1                       pos0 = $byteoffset (naf$t_localuser),
;  654          P 0748  1                       siz0 = naf$s_localuser,
;  655          P 0749  1                       flg = (chg,dup)
;  656            0750  1                       ),
;  657            0751  1
;  658          P 0752  1       nafkey0 : $xabkey (
;  659          P 0753  1                       kref = 0,
;  660          P 0754  1                       pos0 = $byteoffset (naf$t_remname),
;  661          P 0755  1                       siz0 = naf$s_remname,
;  662          P 0756  1                       nxt = nafkey1
;  663            0757  1                       ),
;  664            0758  1
;  665          P 0759  1       nafpro  : $xabpro (
;  666          P 0760  1                       pro = (rwed, rwed),
;  667          P 0761  1                       nxt = nafkey0
;  668            0762  1                       ),
;  669            0763  1
;  670          P 0764  1       naffab  : $fab (                ! FAB for NETUAF
;  671          P 0765  1                       fop = cif,
;  672          P 0766  1                       fac = (get, put, del, upd),
;  673          P 0767  1                       fnm = 'NETUAF',
;  674          P 0768  1                       dnm = '.DAT',
;  675          P 0769  1                       shr = (get, put, del, upd),
;  676          P 0770  1                       org = idx,
;  677          P 0771  1                       rfm = fix,
;  678          P 0772  1                       mrs = naf$c_length,
;  679          P 0773  1                       alq = 10,
;  680          P 0774  1                       deq = 10,
;  681          P 0775  1                       xab = nafpro
;  682            0776  1                       );
;  683            0777  1
;  684            0778  1 !
;  685            0779  1 ! GLOBAL STORAGE:
;  686            0780  1 !
;  687            0781  1 global
;  688            0782  1         faodsc          : block [8, byte],      ! gen'l purpose fao string desc
;  689            0783  1         rabptr          : ref block [rab$c_bln, byte], ! RAB for output
;  690            0784  1         uaf$gq_sysuaff  : block [nsa_s_sysuaff, byte], ! SYSUAF auditing flags
;  691            0785  1         uaf$gl_ctlmsk   : block [2],            ! Control mask for AUTHORIZE
;  692            0786  1         by_account      : long initial (false), ! processing by account
;  693            0787  1         match_token     : vector [naf$s_remname+2, byte], ! Saved match token
;  694            0788  1         match_tokenlen  : long,
```

```
;  695          0789  1          wild_netuser,                          ! wildcard access to proxy entries
;  696          0790  1          call_count      : long initial (0),    ! counter for reprocessing cmd line
;  697          0791  1          tokendsc        : block  [8, byte] preset ( [dsc$b_class]=dsc$k_class_d),
:001 **NEW**    0792  1          cmdlindsc       : block  [DSC$K_D_BLN, byte]
:002 **NEW**    0793  1                            preset([DSC$B_CLASS] = DSC$K_CLASS_D),
;  699-1        0794  1          netuaf_exists,                         ! A self-explanatory flag...
;  700          0795  1          rdb_exists      : long,
;  701          0796  1          rmserr          : long,                ! save rms error codes here
;  702          0797  1          rightslist_modified : byte,            ! RIGHTSLIST modified flag
;  703          0798  1  !
;  704          0799  1  ! Record buffer for file I/O.  Records are generally read into RECBUF,
;  705          0800  1  ! modified, and output.
;  706          0801  1  !
;  707          0802  1          recbuf          : block [uaf$c_length, byte],
;  708          0803  1          netbuf          : block [naf$c_length, byte],
;  709          0804  1  !
;  710          0805  1  ! UAFRAB is global to allow UPDATE_RECORD to modify RAB$W_RSZ.
;  711          0806  1  !
;  712        P 0807  1          uafrab  : $rab (! RAB for work file
;  713        P 0808  1                           krf = 0,
;  714        P 0809  1                           kbf = recbuf [uaf$t_username],
;  715        P 0810  1                           ksz = uaf$s_username,
;  716        P 0811  1                           usz = uaf$c_length,
;  717        P 0812  1                           fab = uaffab
;  718          0813  1                           ),
;  719          0814  1  !
;  720          0815  1  ! RAB for NETUAF Proxy Login File
;  721          0816  1  !
;  722        P 0817  1          nafrab  : $rab (
;  723        P 0818  1                           krf = 0,
;  724        P 0819  1                           kbf = netbuf [naf$t_remname],
;  725        P 0820  1                           ksz = naf$s_remname,
;  726        P 0821  1                           usz = naf$c_length,
;  727        P 0822  1                           rsz = naf$c_length,
;  728        P 0823  1                           rac = key,
;  729        P 0824  1                           fab = naffab
;  730          0825  1                           ),
;  731          0826  1
;  732          0827  1
;  733          0828  1          time_buf        : block [8,byte],              ! current system time
;  734          0829  1          pwd_flag        : long,                ! Password default flag
;  735          0830  1
;  736        P 0831  1          outrab  : $rab (
;  737        P 0832  1                           rac = seq,
;  738        P 0833  1                           rbf = disbuf,
;  739        P 0834  1                           fab = outfab
;  740          0835  1                           ),
;  741          0836  1
;  742          0837  1  !
;  743          0838  1  !        Flag signaling to WILD_USER that a match was found.  This flag
;  744          0839  1  !        is set by the action routine called by WILD_USER.
;  745          0840  1  !
;  746          0841  1          found_match,                   ! found at least one wildcard match
;  747          0842  1
;  748          0843  1  !
;  749          0844  1  ! Wildcard parsing flags set by PARSE_WILD for use in WILD_USER.
;  750          0845  1  !
```

```
;  751     0846  1          uic_flag        : long,
;  752     0847  1          grp_wild        : long,
;  753     0848  1          mem_wild        : long,
;  754     0849  1          str_wild        : long;
;  755     0850  1
;  756     0851  1  global bind
;  757     0852  1          tokenlen        = tokendsc [dsc$w_length] : word,
;  758     0853  1          tokenptr        = tokendsc [dsc$a_pointer],
;  759     0854  1          rec_user_dsc    = uplit (uaf$s_username, recbuf [uaf$t_username]),
;  760     0855  1          rec_encrypt_dsc = uplit (uaf$s_pwd, recbuf [uaf$q_pwd]),
;  761     0856  1          symbol_str      = cstring ('ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789$_');
;  762     0857  1                                    ! valid username characters
;  763     0858  1
;  764     0859  1  !
;  765     0860  1  !       Prompt strings
;  766     0861  1  !
;  767     0862  1
;  768     0863  1  bind
;  769     0864  1
;  770     0865  1      accprmpt     = cstring (%char (lf),'UAF> '),
;  771     0866  1      accprmpt2    = cstring (%char (lf),'_ '),
;  772     0867  1      newmsg20     = cstring ('Do you want to create a new file? ');
;  773     0868  1
;  774     0869  1  !
;  775     0870  1  !       External messages
;  776     0871  1  !
;  777     0872  1
;  778     0873  1  external routine
;  779     0874  1
;  780     0875  1      LIB$SIGNAL;
;  781     0876  1
;  782     0877  1  external literal
;  783     0878  1
;  784     0879  1      UAF$_ADDERR,        UAF$_ADDMSG,        UAF$_BADNODFORM,
;  785     0880  1      UAF$_BADSPC,        UAF$_BADUSR,        UAF$_CLIWARNMSG,
;  786     0881  1      UAF$_CMDTOOLONG,    UAF$_CONERR,        UAF$_COPMSG,
;  787     0882  1      UAF$_CREERR,        UAF$_DEFERR,        UAF$_DEFPWD,
;  788     0883  1      UAF$_DONEMSG,       UAF$_GETERR,        UAF$_HELPERR,
;  789     0884  1      UAF$_INVCMD,        UAF$_INVRSP,        UAF$_INVUSERNAME,
;  790     0885  1      UAF$_KEYNOTFND,     UAF$_KEYNOTUNQ,     UAF$_LSTERR,
;  791     0886  1      UAF$_LSTMSG1,       UAF$_LSTMSG2,       UAF$_MDFYERR,
;  792     0887  1      UAF$_MDFYMSG,       UAF$_NAFADDERR,     UAF$_NAFADDMSG,
;  793     0888  1      UAF$_NAFAEX,        UAF$_NAFCONERR,     UAF$_NAFCREERR,
;  794     0889  1      UAF$_NAFDNE,        UAF$_NAFDONEMSG,    UAF$_NAFNOMODS,
;  795     0890  1      UAF$_NAFUAEERR,     UAF$_NAMETOOBIG,    UAF$_NETLSTMSG,
;  796     0891  1      UAF$_NAOFIL,        UAF$_NAONAF,        UAF$_NOARG,
;  797     0892  1      UAF$_NODEFPWD,      UAF$_NODTOOBIG,     UAF$_NOMODS,
;  798     0893  1      UAF$_NOTUNQ,        UAF$_NOUSERNAME,    UAF$_PREMMSG,
;  799     0894  1      UAF$_PUTERR,        UAF$_RDBDONEMSG,    UAF$_RDBMDFYERR,
;  800     0895  1      UAF$_RDBMDFYERRU,   UAF$_RDBMDFYMSG,    UAF$_RDBNOMODS,
;  801     0896  1      UAF$_REMDEF,        UAF$_REMERR,        UAF$_REMMSG,
;  802     0897  1      UAF$_REMSYS,        UAF$_RENDEF,        UAF$_RENMSG,
;  803     0898  1      UAF$_RENSYS,        UAF$_RONLY,         UAF$_SHOW_ERR,
;  804     0899  1      UAF$_SYSMSG1,       UAF$_SYSMSG2,       UAF$_UAEERR,
;  805     0900  1      UAF$_UICERR,        UAF$_ZISQUAL,       UAF$_ZZPRACREN;
;  806     0901  1
```

```
;  808          0902  1 %sbttl 'start - controlling code'
;  809          0903  1 routine start =
;  810          0904  2 begin
;  811          0905  2
;  812          0906  2 !++
;  813          0907  2 !
;  814          0908  2 ! FUNCTIONAL DESCRIPTION:
;  815          0909  2 !
;  816          0910  2 !       Main procedure of AUTHORIZE.  Call SETUP to initialize
;  817          0911  2 !       all needed files.  Prompt the user for the functions which
;  818          0912  2 !       he/she wants, and call the proper function service routine.
;  819          0913  2 !
;  820          0914  2 ! INPUTS:
;  821          0915  2 !
;  822          0916  2 !       none
;  823          0917  2 !
;  824          0918  2 ! IMPLICIT INPUTS:
;  825          0919  2 !
;  826          0920  2 !       none
;  827          0921  2 !
;  828          0922  2 ! OUTPUTS:
;  829          0923  2 !
;  830          0924  2 !       None
;  831          0925  2 !
;  832          0926  2 ! IMPLICIT OUTPUTS:
;  833          0927  2 !
;  834          0928  2 !       none
;  835          0929  2 !
;  836          0930  2 ! ROUTINE VALUE:
;  837          0931  2 !
;  838          0932  2 !       none
;  839          0933  2 !
;  840          0934  2 ! SIDE EFFECTS:
;  841          0935  2 !
;  842          0936  2 !       none
;  843          0937  2 !--
;  844          0938  2
;  845          0939  2 own
;  846          0940  2         status;
;  847          0941  2
; 854-6         0942  2 rightslist_modified = false;
;  855          0943  2 netuaf_modified = false;
;  856          0944  2 modify_flag = false;                                    ! note no modifications
;  857          0945  2
;  858          0946  2 !
;  859          0947  2 ! Set up terminal I/O
;  860          0948  2 !
;  861          0949  2
;  862          0950  3 if rmsbad (status = $open (fab = infab))
;  863          0951  2 then
;  864          0952  2     signal_stop (.status);
;  865          0953  2
;  866          0954  3 if rmsbad (status = $connect (rab = inrab))
;  867          0955  2 then
;  868          0956  2     signal_stop (.status);
;  869          0957  2
;  870          0958  2 $create (fab = outfab);
```

```
;   871          0959  2 $connect (rab = outrab);
;   872          0960  2
;   873          0961  2 setup ();
;   874          0962  2
;   875          0963  2 !
;   876          0964  2 ! Files have been initialized.  Prompt user for command line
;   877          0965  2 ! and perform requested function.
;   878          0966  2 !
;   879          0967  2
;001 **NEW**     0968  2 if lib$get_foreign (cmdlindsc) and  .cmdlindsc [DSC$W_LENGTH] neq 0
;002 **NEW**     0969  2 then
;003 **NEW**     0970  2     !
;004 **NEW**     0971  2     ! If defined foreign, and there are commands on the line...
;005 **NEW**     0972  2     !
;006 **NEW**     0973  2     begin
;007 **NEW**     0974  3
;008 **NEW**     0975  4          if (status = cli$dcl_parse (cmdlindsc, authorize_commands, 0, LIB$GET_INPUT, %ascid'UAF> '))
;  889-9         0976  3          then
;   890          0977  4             begin
;   891          0978  4             cli$dispatch ();
;   892          0979  4             return true;
;   893          0980  4             end
;   894          0981  3          else
;   895          0982  3              !
;   896          0983  3              ! See if no CLINT exists (Kludge City, USA  37916)
;   897          0984  3              !
;   898          0985  3              if .status eql cli$_noclint
;   899          0986  3              then
;   900          0987  3                  signal_stop (cli$_noclint)
;   901          0988  3              else
;   902          0989  3                  acc$exit ()
;   903          0990  2     end;
;   904          0991  2
;   905          0992  2 while true
;   906          0993  2 do
;   907          0994  3     begin
;   908          0995  3
;   909          0996  3     !
;   910          0997  3     ! Input the command line, taking care of continuations.  Pull off
;   911          0998  3     ! the first token, assuming it is the command name, and look it up
;   912          0999  3     ! in the table of commands.
;   913          1000  3     !
;   914          1001  3
;   915          1002  3     if get_cmd_line ()
;   916          1003  3     then
;001 **NEW**     1004  4          if (status = cli$dcl_parse (cmdlindsc, authorize_commands, 0, LIB$GET_INPUT, %ascid'UAF> '))
;  918-1         1005  3          then
;   919          1006  4              begin
;   920          1007  4              ch$fill (0, uaf$s_flags, uaf$gl_ctlmsk);
;   921          1008  4              cli$dispatch ();
;   922          1009  4              end
;   923          1010  3          else
;   924          1011  3              if .status eql cli$_noclint
;   925          1012  3              then
;   926          1013  3                  signal_stop (cli$_noclint)
;   927          1014  2     end;
;   928          1015  2
```

```
;  929          1016  2 return true;
;  930          1017  2
;  931          1018  1 end;

                                                                .TITLE  UAFMAIN
                                                                .IDENT  \V04-000\

                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                    31 6E 65 6B 6F 74  00000 P.AAB:  .ASCII  \token1\                    ;
                                                       00006         .BLKB   2
                                       00000006        00008 P.AAA:  .LONG   6                          ;
                                       00000000'       0000C         .ADDRESS P.AAB                     ;
                                    32 6E 65 6B 6F 74  00010 P.AAD:  .ASCII  \token2\                    ;
                                                       00016         .BLKB   2
                                       00000006        00018 P.AAC:  .LONG   6                          ;
                                       00000000'       0001C         .ADDRESS P.AAD                     ;
                                       66 65 69 72 62  00020 P.AAF:  .ASCII  \brief\                    ;
                                                       00025         .BLKB   3
                                       00000005        00028 P.AAE:  .LONG   5                          ;
                                       00000000'       0002C         .ADDRESS P.AAF                     ;
                                       6C 6C 75 66     00030 P.AAH:  .ASCII  \full\                     ;
                                       00000004        00034 P.AAG:  .LONG   4                          ;
                                       00000000'       00038         .ADDRESS P.AAH                     ;
    72 65 69 66 69 74 6E 65 64 69 5F 64 64 61          0003C P.AAJ:  .ASCII  \add_identifier\           ;
                                                       0004A         .BLKB   2
                                       0000000E        0004C P.AAI:  .LONG   14                         ;
                                       00000000'       00050         .ADDRESS P.AAJ                     ;
69 66 69 74 6E 65 64 69 5F 65 76 6F 6D 65 72           00054 P.AAL:  .ASCII  \remove_identifier\        ;
                                                          72 65 00063                                   ;
                                                       00065         .BLKB   3                          ;
                                       00000011        00068 P.AAK:  .LONG   17                         ;
                                       00000000'       0006C         .ADDRESS P.AAL                     ;
69 66 69 74 6E 65 64 69 5F 79 66 69 64 6F 6D           00070 P.AAN:  .ASCII  \modify_identifier\        ;
                                                          72 65 0007F                                   ;
                                                       00081         .BLKB   3                          ;
                                       00000011        00084 P.AAM:  .LONG   17                         ;
                                       00000000'       00088         .ADDRESS P.AAN                     ;
6F 73 65 72 2E 73 65 74 75 62 69 72 74 74 61           0008C P.AAO:  .ASCII  \attributes.resource\      ;
                                                          65 63 72 75 0009B                             ;
                                          46 41 55 53 59 53            0009F P.AAP:  .ASCII  \SYSUAF\   ;
                                          46 41 55 54 45 4E            000A5 P.AAQ:  .ASCII  \NETUAF\   ;
                                    64 65 69 66 69 64 6F 6D            000AB P.AAS:  .ASCII  \modified\ ;
                                                       000B3         .BLKB   1                          ;
                                       00000008        000B4 P.AAR:  .LONG   8                          ;
                                       00000000'       000B8         .ADDRESS P.AAS                     ;
                                          64 65 64 64 61               000BC P.AAU:  .ASCII  \added\    ;
                                                       000C1         .BLKB   3                          ;
                                       00000005        000C4 P.AAT:  .LONG   5                          ;
                                       00000000'       000C8         .ADDRESS P.AAU                     ;
                                    64 65 76 6F 6D 65 72               000CC P.AAW:  .ASCII  \removed\  ;
                                                       000D3         .BLKB   1                          ;
                                       00000007        000D4 P.AAV:  .LONG   7                          ;
                                       00000000'       000D8         .ADDRESS P.AAW                     ;
53 41 21 20 53 41 21 20 4C 58 21 3D 44 49 50           000DC P.AAY:  .ASCII  \PID=!XL !AS !AS !AS record !AS on !%D\ ;
53 41 21 20 64 72 6F 63 65 72 20 53 41 21 20           000EB                                            ;
```

```
                                                000FA
                                                00101                   .BLKB   3
                                    00000025    00104 P.AAX:    .LONG   37
                                    00000000'   00108                   .ADDRESS P.AAY
                                      3A  3A    0010C P.ABA:    .ASCII  \::\
                                                0010E                   .BLKB   2
                                    00000002    00110 P.AAZ:    .LONG   2
                                    00000000'   00114                   .ADDRESS P.ABA
                            FFFFFFFF  FFB3B4C0  00118 P.ABB:    .LONG   -5000000, -1
                                          07    00120 P.ABC:    .BYTE   7
                        54  4C  55  41  46  45  44 00121                 .ASCII  \DEFAULT\
                                          04    00128 P.ABD:    .BYTE   4
                                52  45  53  55  00129                   .ASCII  \USER\
                                          09    0012D P.ABE:    .BYTE   9
                    53  45  4C  42  41  54  4C  43 44 0012E             .ASCII  \DCLTABLES\
                                          00    00137 P.ABF:    .BYTE   0
                                                00138                   .BLKB   0
                                          03    00138 P.ABG:    .BYTE   3
                                    4C  43  44  00139                   .ASCII  \DCL\
                                          00    0013C P.ABH:    .BYTE   0
                                                0013D                   .BLKB   0
                                          00    0013D P.ABI:    .BYTE   0
                                                0013E                   .BLKB   0
                                          06    0013E P.ABJ:    .BYTE   6
                            5D  52  45  53  55  5B  0013F                 .ASCII  \[USER]\
                                          00    00145 P.ABK:    .BYTE   0
                                                00146                   .BLKB   0
                                                00146                   .BLKB   2
                                    00000000  00108000  00148 P.ABL:    .LONG   1081344, 0
                                    00000000  00000000  00150 P.ABM:    .LONG   0, 0
                                          06    00158 P.ABN:    .BYTE   6
                                4D  45  54  53  59  53  00159           .ASCII  \SYSTEM\
                                          07    0015F P.ABO:    .BYTE   7
                            52  45  47  41  4E  41  4D 00160             .ASCII  \MANAGER\
                                          09    00167 P.ABP:    .BYTE   9
                        53  45  4C  42  41  54  4C  43 44 00168         .ASCII  \DCLTABLES\
                                          06    00171 P.ABQ:    .BYTE   6
                                4D  45  54  53  59  53  00172           .ASCII  \SYSTEM\
                                          03    00178 P.ABR:    .BYTE   3
                                        4C  43  44  00179               .ASCII  \DCL\
                                          0E    0017C P.ABS:    .BYTE   14
52  45  47  41  4E  41  4D  20  4D  45  54  53  59  53  0017D             .ASCII  \SYSTEM MANAGER\
                                          00    00188 P.ABT:    .BYTE   0
                                                0018C                   .BLKB   0
                                          08    0018C P.ABU:    .BYTE   8
                            5D  52  47  4D  53  59  53  5B 0018D         .ASCII  \[SYSMGR]\
                                          00    00195 P.ABV:    .BYTE   0
                                                00196                   .BLKB   0
                                                00196                   .BLKB   2
                                FFFFFFFF  FFFFFFFF  00198 P.ABW:    .LONG   -1[2]
                                    00000000  00000000  001A0 P.ABX:    .LONG   0, 0
                        54  55  50  4E  49  24  53  59  53 001A8 P.ABY:  .ASCII  \SYS$INPUT\
                    54  55  50  54  55  4F  24  53  59  53 001B1 P.ABZ:  .ASCII  \SYS$OUTPUT\
                    53  49  4C  2E  46  41  55  53  59  53 001BB P.ACA:  .ASCII  \SYSUAF.LIS\
                    53  49  4C  2E  46  41  55  54  45  4E 001C5 P.ACB:  .ASCII  \NETUAF.LIS\
                                46  41  55  53  59  53  001CF P.ACC:    .ASCII  \SYSUAF\
                                        54  41  44  2E  001D5 P.ACD:    .ASCII  \.DAT\
```

```
                                                46  41  55  54  45  4E  001D9 P.ACE:   .ASCII  \NETUAF\                                ;
                                                    54  41  44  2E  001DF P.ACF:   .ASCII  \.DAT\                                 ;
                                                                    001E3          .BLKB   1                                       ;
                                                            00000020    001E4 P.ACG:   .LONG   32                                      ;
                                                            00000000'   001E8          .ADDRESS RECBUF+4                           ;
                                                            00000008    001EC P.ACH:   .LONG   8                                       ;
                                                            00000000'   001F0          .ADDRESS RECBUF+340                         ;
                                                                26      001F4 P.ACI:   .BYTE   38                                      ;
4F  4E  4D  4C  4B  4A  49  48  47  46  45  44  43  42  41   001F5          .ASCII  \ABCDEFGHIJKLMNOPQRSTUVWXYZ01234567890$_\           ;
33  32  31  30  5A  59  58  57  56  55  54  53  52  51  50   00204                                                                     ;
                        5F  24  39  38  37  36  35  34      00213                                                                     ;
                                                                06      0021B P.ACJ:   .BYTE   6                                       ;
                                                    20  3E  46  41  55  0A  0021C          .ASCII  <10>\UAF> \                      ;
                                                                03      00222 P.ACK:   .BYTE   3                                       ;
                                                            20  5F  0A  00223          .ASCII  <10>\_ \                         ;
                                                                22      00226 P.ACL:   .BYTE   34                                      ;
20  6F  74  20  74  6E  61  77  20  75  6F  79  20  6F  44   00227          .ASCII  \Do you want to create a new file? \              ;
69  66  20  77  65  6E  20  61  20  65  74  61  65  72  63   00236                                                                     ;
                                        20  3F  65  6C      00245                                                                  ;
                                                            00249          .BLKB   3                                               ;
                        00  00  00  20  3E  46  41  55      0024C P.ACN:   .ASCII  \UAF> \<0><0><0>                                ;
                                                    010E0005    00254 P.ACM:   .LONG   17694725                                    ;
                                                    00000000'   00258          .ADDRESS P.ACN                                      ;
                        00  00  00  20  3E  46  41  55      0025C P.ACP:   .ASCII  \UAF> \<0><0><0>                                ;
                                                    010E0005    00264 P.ACO:   .LONG   17694725                                    ;
                                                    00000000'   00268          .ADDRESS P.ACP                                      ;

                                                               .PSECT  $OWN$,NOEXE,2

    0013  00000 SD_ATTRIBRESOURCE:
                               .WORD   19                                                                                          ;
  01  0E  00002                .BYTE   14, 1                                                                                       ;
00000000' 00004                .ADDRESS P.AAO                                                                                      ;
          00008 USERNAME_BUF:
                               .BLKB   32
          00028 PCB_STS:       .BLKB   4
          0002C PID:           .BLKB   4
00000000  00030 USERNAME_DSC:
                               .LONG   0                                                                                           ;
00000000' 00034                .ADDRESS USERNAME_BUF                                                                               ;
          00038 RECNAME_DSC:
                               .BLKB   8
00000006  00040 FILE_DSC:
                               .LONG   6                                                                                           ;
          00044                .BLKB   4
          00048 MOD_DEFAULT:
                               .BLKB   4
          0004C MODIFY_FLAG:
                               .BLKB   4
          00050 NETUAF_MODIFIED:
                               .BLKB   4
       00 00054 RENAME_PH2:
                               .BYTE   0                                                                                           ;
          00055                .BLKB   3
          00058 OLDUSERLEN:
                               .BLKB   4
          0005C OLDUSERNAME:
```

```
                                        .BLKB   32
                        0007C NEWUSERLEN:
                                        .BLKB   4
                        00080 NEWUSERNAME:
                                        .BLKB   32
                        000A0 CMDBUF:   .BLKB   1024
                        004A0 DEFAULT_SIZE:
                                        .BLKB   2
                        004A2           .BLKB   2
                        004A4 DEFAULT_RECORD:
                                        .BLKB   1412
                        00A28 PWDDSC:   .BLKB   8
                        00A30 INSIZE:   .BLKB   4
                        00A34 BRIEF_FLAG:
                                        .BLKB   4
                        00A38 FULL_FLAG:
                                        .BLKB   4
                        00A3C HEADER_FLAG:
                                        .BLKB   4
              03        00A40 INFAB:    .BYTE   3
              50        00A41           .BYTE   80
            0000        00A42           .WORD   0
        00000000        00A44           .LONG   0
        00000000        00A48           .LONG   0
        00000000        00A4C           .LONG   0
        00000000        00A50           .LONG   0
            0000        00A54           .WORD   0
              03        00A56           .BYTE   3
              00        00A57           .BYTE   0
        00000000        00A58           .LONG   0
              00        00A5C           .BYTE   0
              00        00A5D           .BYTE   0
              02        00A5E           .BYTE   2
              02        00A5F           .BYTE   2
        00000000        00A60           .LONG   0
        00000000        00A64           .LONG   0
        00000000        00A68           .LONG   0
        00000000'       00A6C           .ADDRESS P.ABY
        00000000        00A70           .LONG   0
              09        00A74           .BYTE   9
              00        00A75           .BYTE   0
            0000        00A76           .WORD   0
        00000000        00A78           .LONG   0
            0000        00A7C           .WORD   0
              00        00A7E           .BYTE   0
              00        00A7F           .BYTE   0
        00000000        00A80           .LONG   0
        00000000        00A84           .LONG   0
            0000        00A88           .WORD   0
              00        00A8A           .BYTE   0
              00        00A8B           .BYTE   0
        00000000        00A8C           .LONG   0
              01        00A90 INRAB:    .BYTE   1
              44        00A91           .BYTE   68
            0000        00A92           .WORD   0
        40000000        00A94           .LONG   1073741824
        00000000        00A98           .LONG   0
```

```
00000000  00A9C          .LONG   0
    0000# 00AA0          .WORD   0[3]
     0000 00AA6          .WORD   0
00000000  00AA8          .LONG   0
     0000 00AAC          .WORD   0
       00 00AAE          .BYTE   0
       00 00AAF          .BYTE   0
     0000 00AB0          .WORD   0
     0000 00AB2          .WORD   0
00000000  00AB4          .LONG   0
00000000  00AB8          .LONG   0
00000000  00ABC          .LONG   0
00000000  00AC0          .LONG   0
       00 00AC4          .BYTE   0
       00 00AC5          .BYTE   0
       00 00AC6          .BYTE   0
       00 00AC7          .BYTE   0
00000000  00AC8          .LONG   0
00000000' 00ACC          .ADDRESS INFAB
00000000  00AD0          .LONG   0
       03 00AD4 OUTFAB:  .BYTE   3
       50 00AD5          .BYTE   80
     0000 00AD6          .WORD   0
00000000  00AD8          .LONG   0
00000000  00ADC          .LONG   0
00000000  00AE0          .LONG   0
00000000  00AE4          .LONG   0
     0000 00AE8          .WORD   0
       01 00AEA          .BYTE   1
       00 00AEB          .BYTE   0
00000000  00AEC          .LONG   0
       00 00AF0          .BYTE   0
       00 00AF1          .BYTE   0
       02 00AF2          .BYTE   2
       02 00AF3          .BYTE   2
00000000  00AF4          .LONG   0
00000000  00AF8          .LONG   0
00000000  00AFC          .LONG   0
00000000' 00B00          .ADDRESS P.ABZ
00000000  00B04          .LONG   0
       0A 00B08          .BYTE   10
       00 00B09          .BYTE   0
     0000 00B0A          .WORD   0
00000000  00B0C          .LONG   0
     0000 00B10          .WORD   0
       00 00B12          .BYTE   0
       00 00B13          .BYTE   0
00000000  00B14          .LONG   0
00000000  00B18          .LONG   0
     0000 00B1C          .WORD   0
       00 00B1E          .BYTE   0
       00 00B1F          .BYTE   0
00000000  00B20          .LONG   0
       02 00B24 LSTNAM:  .BYTE   2
       60 00B25          .BYTE   96
       00 00B26          .BYTE   0
       00 00B27          .BYTE   0
```

```
00000000  00B28           .LONG   0
       00  00B2C           .BYTE   0
       00  00B2D           .BYTE   0
       00  00B2E           .BYTE   0
       00  00B2F           .BYTE   0
00000000  00B30           .LONG   0
00000000  00B34           .LONG   0
    0000# 00B38           .WORD   0[8]
    0000# 00B48           .WORD   0[3]
    0000# 00B4E           .WORD   0[3]
00000000  00B54           .LONG   0
00000000  00B58           .LONG   0
       00  00B5C           .BYTE   0
       00  00B5D           .BYTE   0
       00  00B5E           .BYTE   0
       00  00B5F           .BYTE   0
       00  00B60           .BYTE   0
       00  00B61           .BYTE   0
      00#  00B62           .BYTE   0[2]
00000000  00B64           .LONG   0
00000000  00B68           .LONG   0
00000000  00B6C           .LONG   0
00000000  00B70           .LONG   0
00000000  00B74           .LONG   0
00000000  00B78           .LONG   0
00000000# 00B7C           .LONG   0[2]
       13  00B84 LSTPRO:  .BYTE   19
       58  00B85           .BYTE   88
     0000  00B86           .WORD   0
00000000  00B88           .LONG   0
     FC44  00B8C           .WORD   -956
       00  00B8E           .BYTE   0
       00  00B8F           .BYTE   0
0000 0000  00B90           .WORD   0, 0
       00  00B94           .BYTE   0
       00  00B95           .BYTE   0
     0000  00B96           .WORD   0
00000000  00B98           .LONG   0
00000000  00B9C           .LONG   0
     0000  00BA0           .WORD   0
     0000  00BA2           .WORD   0
00000000  00BA4           .LONG   0
00000000  00BA8           .LONG   0
          00BAC           .BLKB   48
       03  00BDC LSTFAB:  .BYTE   3
       50  00BDD           .BYTE   80
     0000  00BDE           .WORD   0
00000000  00BE0           .LONG   0
00000000  00BE4           .LONG   0
00000000  00BE8           .LONG   0
00000000  00BEC           .LONG   0
     0000  00BF0           .WORD   0
       01  00BF2           .BYTE   1
       20  00BF3           .BYTE   32
00000000  00BF4           .LONG   0
       00  00BF8           .BYTE   0
       00  00BF9           .BYTE   0
```

```
                02  00BFA                  .BYTE   2
                02  00BFB                  .BYTE   2
00000000'       00  00BFC                  .LONG   0
00000000'       00  00C00                  .ADDRESS LSTPRO
00000000'       00  00C04                  .ADDRESS LSTNAM
00000000'       00  00C08                  .ADDRESS P.ACA
00000000        00  00C0C                  .LONG   0
                0A  00C10                  .BYTE   10
                00  00C11                  .BYTE   0
              0084  00C12                  .WORD   132
00000000        00  00C14                  .LONG   0
              0000  00C18                  .WORD   0
                00  00C1A                  .BYTE   0
                00  00C1B                  .BYTE   0
00000000        00  00C1C                  .LONG   0
00000000        00  00C20                  .LONG   0
              0000  00C24                  .WORD   0
                00  00C26                  .BYTE   0
                00  00C27                  .BYTE   0
00000000        00  00C28                  .LONG   0
                01  00C2C  LSTRAB:         .BYTE   1
                44  00C2D                  .BYTE   68
              0000  00C2E                  .WORD   0
00000000        00  00C30                  .LONG   0
00000000        00  00C34                  .LONG   0
00000000        00  00C38                  .LONG   0
              0000# 00C3C                  .WORD   0[3]
              0000  00C42                  .WORD   0
00000000        00  00C44                  .LONG   0
              0000  00C48                  .WORD   0
                00  00C4A                  .BYTE   0
                00  00C4B                  .BYTE   0
              0000  00C4C                  .WORD   0
              0000  00C4E                  .WORD   0
00000000        00  00C50                  .LONG   0
00000000'       00  00C54                  .ADDRESS DISBUF
00000000        00  00C58                  .LONG   0
00000000        00  00C5C                  .LONG   0
                00  00C60                  .BYTE   0
                00  00C61                  .BYTE   0
                00  00C62                  .BYTE   0
                00  00C63                  .BYTE   0
00000000        00  00C64                  .LONG   0
00000000'       00  00C68                  .ADDRESS LSTFAB
00000000        00  00C6C                  .LONG   0
                02  00C70  NLSTNAM:        .BYTE   2
                60  00C71                  .BYTE   96
                00  00C72                  .BYTE   0
                00  00C73                  .BYTE   0
00000000        00  00C74                  .LONG   0
                00  00C78                  .BYTE   0
                00  00C79                  .BYTE   0
                00  00C7A                  .BYTE   0
                00  00C7B                  .BYTE   0
00000000        00  00C7C                  .LONG   0
00000000        00  00C80                  .LONG   0
              0000# 00C84                  .WORD   0[8]
```

```
                0000#  00C94                .WORD   0[3]
                0000#  00C9A                .WORD   0[3]
            00000000   00CA0                .LONG   0
            00000000   00CA4                .LONG   0
                  00   00CA8                .BYTE   0
                  00   00CA9                .BYTE   0
                  00   00CAA                .BYTE   0
                  00   00CAB                .BYTE   0
                  00   00CAC                .BYTE   0
                  00   00CAD                .BYTE   0
                 00#   00CAE                .BYTE   0[2]
            00000000   00CB0                .LONG   0
            00000000   00CB4                .LONG   0
            00000000   00CB8                .LONG   0
            00000000   00CBC                .LONG   0
            00000000   00CC0                .LONG   0
            00000000   00CC4                .LONG   0
            00000000#  00CC8                .LONG   0[2]
                  13   00CD0 NLSTPRO:       .BYTE   19
                  58   00CD1                .BYTE   88
                0000   00CD2                .WORD   0
            00000000   00CD4                .LONG   0
                FC44   00CD8                .WORD   -956
                  00   00CDA                .BYTE   0
                  00   00CDB                .BYTE   0
            0000  0000 00CDC                .WORD   0, 0
                  00   00CE0                .BYTE   0
                  00   00CE1                .BYTE   0
                0000   00CE2                .WORD   0
            00000000   00CE4                .LONG   0
            00000000   00CE8                .LONG   0
                0000   00CEC                .WORD   0
                0000   00CEE                .WORD   0
            00000000   00CF0                .LONG   0
            00000000   00CF4                .LONG   0
                       00CF8                .BLKB   48
                  03   00D28 NLSTFAB:       .BYTE   3
                  50   00D29                .BYTE   80
                0000   00D2A                .WORD   0
            00000000   00D2C                .LONG   0
            00000000   00D30                .LONG   0
            00000000   00D34                .LONG   0
            00000000   00D38                .LONG   0
                0000   00D3C                .WORD   0
                  01   00D3E                .BYTE   1
                  20   00D3F                .BYTE   32
            00000000   00D40                .LONG   0
                  00   00D44                .BYTE   0
                  00   00D45                .BYTE   0
                  02   00D46                .BYTE   2
                  02   00D47                .BYTE   2
            00000000   00D48                .LONG   0
            00000000'  00D4C                .ADDRESS NLSTPRO
            00000000'  00D50                .ADDRESS NLSTNAM
            00000000'  00D54                .ADDRESS P.ACB
            00000000   00D58                .LONG   0
                  0A   00D5C                .BYTE   10
```

```
                00  00D5D                .BYTE   0                                                       ;
              0084  00D5E                .WORD   132                                                     ;
          00000000  00D60                .LONG   0                                                       ;
              0000  00D64                .WORD   0                                                       ;
                00  00D66                .BYTE   0                                                       ;
                00  00D67                .BYTE   0                                                       ;
          00000000  00D68                .LONG   0                                                       ;
          00000000  00D6C                .LONG   0                                                       ;
              0000  00D70                .WORD   0                                                       ;
                00  00D72                .BYTE   0                                                       ;
                00  00D73                .BYTE   0                                                       ;
          00000000  00D74                .LONG   0                                                       ;
                01  00D78 NLSTRAB:       .BYTE   1                                                       ;
                44  00D79                .BYTE   68                                                      ;
              0000  00D7A                .WORD   0                                                       ;
          00000000  00D7C                .LONG   0                                                       ;
          00000000  00D80                .LONG   0                                                       ;
          00000000  00D84                .LONG   0                                                       ;
             0000#  00D88                .WORD   0[3]                                                    ;
              0000  00D8E                .WORD   0                                                       ;
          00000000  00D90                .LONG   0                                                       ;
              0000  00D94                .WORD   0                                                       ;
                00  00D96                .BYTE   0                                                       ;
                00  00D97                .BYTE   0                                                       ;
              0000  00D98                .WORD   0                                                       ;
              0000  00D9A                .WORD   0                                                       ;
          00000000  00D9C                .LONG   0                                                       ;
          00000000' 00DA0                .ADDRESS DISBUF                                                 ;
          00000000  00DA4                .LONG   0                                                       ;
          00000000  00DA8                .LONG   0                                                       ;
                00  00DAC                .BYTE   0                                                       ;
                00  00DAD                .BYTE   0                                                       ;
                00  00DAE                .BYTE   0                                                       ;
                00  00DAF                .BYTE   0                                                       ;
          00000000  00DB0                .LONG   0                                                       ;
          00000000' 00DB4                .ADDRESS NLSTFAB                                                ;
          00000000  00DB8                .LONG   0                                                       ;
                15  00DBC UAFKEY2:       .BYTE   21                                                      ;
                4C  00DBD                .BYTE   76                                                      ;
              0000  00DBE                .WORD   0                                                       ;
          00000000  00DC0                .LONG   0                                                       ;
                00  00DC4                .BYTE   0                                                       ;
                00  00DC5                .BYTE   0                                                       ;
                00  00DC6                .BYTE   0                                                       ;
                00  00DC7                .BYTE   0                                                       ;
                00  00DC8                .BYTE   0                                                       ;
                00  00DC9                .BYTE   0                                                       ;
          00000000  00DCA                .LONG   0                                                       ;
                03  00DCE                .BYTE   3                                                       ;
                02  00DCF                .BYTE   2                                                       ;
                00  00DD0                .BYTE   0                                                       ;
                00  00DD1                .BYTE   0                                                       ;
                00  00DD2                .BYTE   0                                                       ;
                02  00DD3                .BYTE   2                                                       ;
              0000  00DD4                .WORD   0                                                       ;
              0000  00DD6                .WORD   0                                                       ;
              0000  00DD8                .WORD   0                                                       ;
```

```
                0024  00DDA           .WORD   36
                0000  00DDC           .WORD   0
                0000  00DDE           .WORD   0
                0000  00DE0           .WORD   0
                0000  00DE2           .WORD   0
                0000  00DE4           .WORD   0
                0000  00DE6           .WORD   0
                0000  00DE8           .WORD   0
                  02  00DEA           .BYTE   2
                  00  00DEB           .BYTE   0
                  00  00DEC           .BYTE   0
                  00  00DED           .BYTE   0
                  00  00DEE           .BYTE   0
                  00  00DEF           .BYTE   0
                  00  00DF0           .BYTE   0
                  00  00DF1           .BYTE   0
                0000  00DF2           .WORD   0
            00000000  00DF4           .LONG   0
            00000000  00DF8           .LONG   0
                  00  00DFC           .BYTE   0
                  00  00DFD           .BYTE   0
                  00  00DFE           .BYTE   0
                  00  00DFF           .BYTE   0
                  00  00E00           .BYTE   0
                  00  00E01           .BYTE   0
                  00  00E02           .BYTE   0
                  00  00E03           .BYTE   0
                  00  00E04           .BYTE   0
                  00  00E05           .BYTE   0
                      00E06           .BLKB   2
                  15  00E08 UAFKEY1:  .BYTE   21
                  4C  00E09           .BYTE   76
                0000  00E0A           .WORD   0
            00000000' 00E0C           .ADDRESS UAFKEY2
                  00  00E10           .BYTE   0
                  00  00E11           .BYTE   0
                  00  00E12           .BYTE   0
                  00  00E13           .BYTE   0
                  00  00E14           .BYTE   0
                  00  00E15           .BYTE   0
            00000000  00E16           .LONG   0
                  03  00E1A           .BYTE   3
                  04  00E1B           .BYTE   4
                  00  00E1C           .BYTE   0
                  00  00E1D           .BYTE   0
                  00  00E1E           .BYTE   0
                  01  00E1F           .BYTE   1
                0000  00E20           .WORD   0
                0000  00E22           .WORD   0
                0000  00E24           .WORD   0
                0024  00E26           .WORD   36
                0000  00E28           .WORD   0
                0000  00E2A           .WORD   0
                0000  00E2C           .WORD   0
                0000  00E2E           .WORD   0
                0000  00E30           .WORD   0
                0000  00E32           .WORD   0
```

```
                0000  00E34          .WORD   0                                                       ;
                  04  00E36          .BYTE   4                                                       ;
                  00  00E37          .BYTE   0                                                       ;
                  00  00E38          .BYTE   0                                                       ;
                  00  00E39          .BYTE   0                                                       ;
                  00  00E3A          .BYTE   0                                                       ;
                  00  00E3B          .BYTE   0                                                       ;
                  00  00E3C          .BYTE   0                                                       ;
                  00  00E3D          .BYTE   0                                                       ;
                0000  00E3E          .WORD   0                                                       ;
            00000000  00E40          .LONG   0                                                       ;
            00000000  00E44          .LONG   0                                                       ;
                  00  00E48          .BYTE   0                                                       ;
                  00  00E49          .BYTE   0                                                       ;
                  00  00E4A          .BYTE   0                                                       ;
                  00  00E4B          .BYTE   0                                                       ;
                  00  00E4C          .BYTE   0                                                       ;
                  00  00E4D          .BYTE   0                                                       ;
                  00  00E4E          .BYTE   0                                                       ;
                  00  00E4F          .BYTE   0                                                       ;
                  00  00E50          .BYTE   0                                                       ;
                  00  00E51          .BYTE   0                                                       ;
                      00E52          .BLKB   2                                                       ;
                  15  00E54  UAFKEY0: .BYTE   21                                                      ;
                  4C  00E55          .BYTE   76                                                      ;
                0000  00E56          .WORD   0                                                       ;
            00000000' 00E58          .ADDRESS UAFKEY1                                                ;
                  00  00E5C          .BYTE   0                                                       ;
                  00  00E5D          .BYTE   0                                                       ;
                  00  00E5E          .BYTE   0                                                       ;
                  00  00E5F          .BYTE   0                                                       ;
                  00  00E60          .BYTE   0                                                       ;
                  00  00E61          .BYTE   0                                                       ;
            00000000  00E62          .LONG   0                                                       ;
                  00  00E66          .BYTE   0                                                       ;
                  00  00E67          .BYTE   0                                                       ;
                  00  00E68          .BYTE   0                                                       ;
                  00  00E69          .BYTE   0                                                       ;
                  00  00E6A          .BYTE   0                                                       ;
                  00  00E6B          .BYTE   0                                                       ;
                0000  00E6C          .WORD   0                                                       ;
                0000  00E6E          .WORD   0                                                       ;
                0000  00E70          .WORD   0                                                       ;
                0004  00E72          .WORD   4                                                       ;
                0000  00E74          .WORD   0                                                       ;
                0000  00E76          .WORD   0                                                       ;
                0000  00E78          .WORD   0                                                       ;
                0000  00E7A          .WORD   0                                                       ;
                0000  00E7C          .WORD   0                                                       ;
                0000  00E7E          .WORD   0                                                       ;
                0000  00E80          .WORD   0                                                       ;
                  20  00E82          .BYTE   32                                                      ;
                  00  00E83          .BYTE   0                                                       ;
                  00  00E84          .BYTE   0                                                       ;
                  00  00E85          .BYTE   0                                                       ;
                  00  00E86          .BYTE   0                                                       ;
                  00  00E87          .BYTE   0                                                       ;
```

```
                00  00E88          .BYTE   0                                       ;
                00  00E89          .BYTE   0                                       ;
              0000  00E8A          .WORD   0                                       ;
          00000000  00E8C          .LONG   0                                       ;
          00000000  00E90          .LONG   0                                       ;
                00  00E94          .BYTE   0                                       ;
                00  00E95          .BYTE   0                                       ;
                00  00E96          .BYTE   0                                       ;
                00  00E97          .BYTE   0                                       ;
                00  00E98          .BYTE   0                                       ;
                00  00E99          .BYTE   0                                       ;
                00  00E9A          .BYTE   0                                       ;
                00  00E9B          .BYTE   0                                       ;
                00  00E9C          .BYTE   0                                       ;
                00  00E9D          .BYTE   0                                       ;
                    00E9E          .BLKB   2                                       ;
                13  00EA0 UAFPRO:  .BYTE   19                                      ;
                58  00EA1          .BYTE   88                                      ;
              0000  00EA2          .WORD   0                                       ;
          00000000' 00EA4          .ADDRESS UAFKEY0                                ;
              FF00  00EA8          .WORD   -256                                    ;
                00  00EAA          .BYTE   0                                       ;
                00  00EAB          .BYTE   0                                       ;
        0000  0000  00EAC          .WORD   0, 0                                    ;
                00  00EB0          .BYTE   0                                       ;
                00  00EB1          .BYTE   0                                       ;
              0000  00EB2          .WORD   0                                       ;
          00000000  00EB4          .LONG   0                                       ;
          00000000  00EB8          .LONG   0                                       ;
              0000  00EBC          .WORD   0                                       ;
              0000  00EBE          .WORD   0                                       ;
          00000000  00EC0          .LONG   0                                       ;
          00000000  00EC4          .LONG   0                                       ;
                    00EC8          .BLKB   48                                      ;
                03  00EF8 UAFFAB:  .BYTE   3                                       ;
                50  00EF9          .BYTE   80                                      ;
              0000  00EFA          .WORD   0                                       ;
          02000000  00EFC          .LONG   33554432                                ;
          00000000  00F00          .LONG   0                                       ;
          00000000  00F04          .LONG   0                                       ;
          0000000A  00F08          .LONG   10                                      ;
              000A  00F0C          .WORD   10                                      ;
                0F  00F0E          .BYTE   15                                      ;
                0F  00F0F          .BYTE   15                                      ;
          00000000  00F10          .LONG   0                                       ;
                00  00F14          .BYTE   0                                       ;
                20  00F15          .BYTE   32                                      ;
                00  00F16          .BYTE   0                                       ;
                02  00F17          .BYTE   2                                       ;
          00000000  00F18          .LONG   0                                       ;
          00000000' 00F1C          .ADDRESS UAFPRO                                 ;
          00000000  00F20          .LONG   0                                       ;
          00000000' 00F24          .ADDRESS P.ACC                                  ;
          00000000' 00F28          .ADDRESS P.ACD                                  ;
                06  00F2C          .BYTE   6                                       ;
                04  00F2D          .BYTE   4                                       ;
              0584  00F2E          .WORD   1412                                    ;
```

```
00000000  00F30          .LONG   0
    0000  00F34          .WORD   0
      00  00F36          .BYTE   0
      00  00F37          .BYTE   0
00000000  00F38          .LONG   0
00000000  00F3C          .LONG   0
    0000  00F40          .WORD   0
      00  00F42          .BYTE   0
      00  00F43          .BYTE   0
00000000  00F44          .LONG   0
      15  00F48 NAFKEY1: .BYTE   21
      4C  00F49          .BYTE   76
    0000  00F4A          .WORD   0
00000000  00F4C          .LONG   0
      00  00F50          .BYTE   0
      00  00F51          .BYTE   0
      00  00F52          .BYTE   0
      00  00F53          .BYTE   0
      00  00F54          .BYTE   0
      00  00F55          .BYTE   0
00000000  00F56          .LONG   0
      03  00F5A          .BYTE   3
      00  00F5B          .BYTE   0
      00  00F5C          .BYTE   0
      00  00F5D          .BYTE   0
      00  00F5E          .BYTE   0
      01  00F5F          .BYTE   1
    0000  00F60          .WORD   0
    0000  00F62          .WORD   0
    0000  00F64          .WORD   0
    0040  00F66          .WORD   64
    0000  00F68          .WORD   0
    0000  00F6A          .WORD   0
    0000  00F6C          .WORD   0
    0000  00F6E          .WORD   0
    0000  00F70          .WORD   0
    0000  00F72          .WORD   0
    0000  00F74          .WORD   0
      20  00F76          .BYTE   32
      00  00F77          .BYTE   0
      00  00F78          .BYTE   0
      00  00F79          .BYTE   0
      00  00F7A          .BYTE   0
      00  00F7B          .BYTE   0
      00  00F7C          .BYTE   0
      00  00F7D          .BYTE   0
    0000  00F7E          .WORD   0
00000000  00F80          .LONG   0
00000000  00F84          .LONG   0
      00  00F88          .BYTE   0
      00  00F89          .BYTE   0
      00  00F8A          .BYTE   0
      00  00F8B          .BYTE   0
      00  00F8C          .BYTE   0
      00  00F8D          .BYTE   0
      00  00F8E          .BYTE   0
      00  00F8F          .BYTE   0
```

```
                    00  00F90          .BYTE   0                                                       ;
                    00  00F91          .BYTE   0                                                       ;
                        00F92          .BLKB   2                                                       ;
                    15  00F94 NAFKEY0: .BYTE   21                                                      ;
                    4C  00F95          .BYTE   76                                                      ;
                  0000  00F96          .WORD   0                                                       ;
            00000000'   00F98          .ADDRESS NAFKEY1                                                ;
                    00  00F9C          .BYTE   0                                                       ;
                    00  00F9D          .BYTE   0                                                       ;
                    00  00F9E          .BYTE   0                                                       ;
                    00  00F9F          .BYTE   0                                                       ;
                    00  00FA0          .BYTE   0                                                       ;
                    00  00FA1          .BYTE   0                                                       ;
            00000000    00FA2          .LONG   0                                                       ;
                    00  00FA6          .BYTE   0                                                       ;
                    00  00FA7          .BYTE   0                                                       ;
                    00  00FA8          .BYTE   0                                                       ;
                    00  00FA9          .BYTE   0                                                       ;
                    00  00FAA          .BYTE   0                                                       ;
                    00  00FAB          .BYTE   0                                                       ;
                  0000  00FAC          .WORD   0                                                       ;
                  0000  00FAE          .WORD   0                                                       ;
                  0000  00FB0          .WORD   0                                                       ;
                  0000  00FB2          .WORD   0                                                       ;
                  0000  00FB4          .WORD   0                                                       ;
                  0000  00FB6          .WORD   0                                                       ;
                  0000  00FB8          .WORD   0                                                       ;
                  0000  00FBA          .WORD   0                                                       ;
                  0000  00FBC          .WORD   0                                                       ;
                  0000  00FBE          .WORD   0                                                       ;
                  0000  00FC0          .WORD   0                                                       ;
                    40  00FC2          .BYTE   64                                                      ;
                    00  00FC3          .BYTE   0                                                       ;
                    00  00FC4          .BYTE   0                                                       ;
                    00  00FC5          .BYTE   0                                                       ;
                    00  00FC6          .BYTE   0                                                       ;
                    00  00FC7          .BYTE   0                                                       ;
                    00  00FC8          .BYTE   0                                                       ;
                    00  00FC9          .BYTE   0                                                       ;
                  0000  00FCA          .WORD   0                                                       ;
            00000000    00FCC          .LONG   0                                                       ;
            00000000    00FD0          .LONG   0                                                       ;
                    00  00FD4          .BYTE   0                                                       ;
                    00  00FD5          .BYTE   0                                                       ;
                    00  00FD6          .BYTE   0                                                       ;
                    00  00FD7          .BYTE   0                                                       ;
                    00  00FD8          .BYTE   0                                                       ;
                    00  00FD9          .BYTE   0                                                       ;
                    00  00FDA          .BYTE   0                                                       ;
                    00  00FDB          .BYTE   0                                                       ;
                    00  00FDC          .BYTE   0                                                       ;
                    00  00FDD          .BYTE   0                                                       ;
                        00FDE          .BLKB   2                                                       ;
                    13  00FE0 NAFPRO:  .BYTE   19                                                      ;
                    58  00FE1          .BYTE   88                                                      ;
                  0000  00FE2          .WORD   0                                                       ;
            00000000'   00FE4          .ADDRESS NAFKEY0                                                ;
```

```
                FF00  00FE8          .WORD   -256                                    ;
                  00  00FEA          .BYTE   0                                       ;
                  00  00FEB          .BYTE   0                                       ;
0000            0000  00FEC          .WORD   0, 0                                    ;
                  00  00FF0          .BYTE   0                                       ;
                  00  00FF1          .BYTE   0                                       ;
                0000  00FF2          .WORD   0                                       ;
          00000000  00FF4          .LONG   0                                       ;
          00000000  00FF8          .LONG   0                                       ;
                0000  00FFC          .WORD   0                                       ;
                0000  00FFE          .WORD   0                                       ;
          00000000  01000          .LONG   0                                       ;
          00000000  01004          .LONG   0                                       ;
                    01008          .BLKB   48                                      ;
                  03  01038 NAFFAB:  .BYTE   3                                      ;
                  50  01039          .BYTE   80                                      ;
                0000  0103A          .WORD   0                                       ;
          02000000  0103C          .LONG   33554432                                ;
          00000000  01040          .LONG   0                                       ;
          00000000  01044          .LONG   0                                       ;
          0000000A  01048          .LONG   10                                      ;
                000A  0104C          .WORD   10                                      ;
                  0F  0104E          .BYTE   15                                      ;
                  0F  0104F          .BYTE   15                                      ;
          00000000  01050          .LONG   0                                       ;
                  00  01054          .BYTE   0                                       ;
                  20  01055          .BYTE   32                                      ;
                  00  01056          .BYTE   0                                       ;
                  01  01057          .BYTE   1                                       ;
          00000000  01058          .LONG   0                                       ;
          00000000' 0105C          .ADDRESS NAFPRO                                 ;
          00000000  01060          .LONG   0                                       ;
          00000000' 01064          .ADDRESS P.ACE                                  ;
          00000000' 01068          .ADDRESS P.ACF                                  ;
                  06  0106C          .BYTE   6                                       ;
                  04  0106D          .BYTE   4                                       ;
                0064  0106E          .WORD   100                                     ;
          00000000  01070          .LONG   0                                       ;
                0000  01074          .WORD   0                                       ;
                  00  01076          .BYTE   0                                       ;
                  00  01077          .BYTE   0                                       ;
          00000000  01078          .LONG   0                                       ;
          00000000  0107C          .LONG   0                                       ;
                0000  01080          .WORD   0                                       ;
                  00  01082          .BYTE   0                                       ;
                  00  01083          .BYTE   0                                       ;
          00000000  01084          .LONG   0                                       ;
                    01088 STATUS:  .BLKB   4                                       ;

                                   .PSECT  $GLOBAL$,NOEXE,2

                    00000 DISBUF::.BLKB   132
          00000084  00084 DISDSC::.LONG   132                                       ;
          00000000' 00088          .ADDRESS DISBUF                                 ;
                    0008C FAODSC::.BLKB   8
                    00094 RABPTR::.BLKB   4
                    00098 UAF$GQ_SYSUAFF::
```

```
                                .BLKB   8
                000A0 UAF$GL_CTLMSK::
                                .BLKB   8
00000000        000A8 BY_ACCOUNT::
                                .LONG   0                                       ;
                000AC MATCH_TOKEN::
                                .BLKB   66
                000EE           .BLKB   2
                000F0 MATCH_TOKENLEN::
                                .BLKB   4
                000F4 WILD_NETUSER::
                                .BLKB   4
00000000        000F8 CALL_COUNT::
                                .LONG   0                                       ;
      00#       000FC TOKENDSC::
                                .BYTE   0[3]                                    ;
      02        000FF           .BYTE   2                                       ;
                00100           .BLKB   4
      00#       00104 CMDLINDSC::
                                .BYTE   0[3]                                    ;
      02        00107           .BYTE   2                                       ;
                00108           .BLKB   4
                0010C NETUAF_EXISTS::
                                .BLKB   4
                00110 RDB_EXISTS::
                                .BLKB   4
                00114 RMSERR::  .BLKB   4
                00118 RIGHTSLIST_MODIFIED::
                                .BLKB   1
                00119           .BLKB   3
                0011C RECBUF::  .BLKB   1412
                006A0 NETBUF::  .BLKB   100
      01        00704 UAFRAB::  .BYTE   1                                       ;
      44        00705           .BYTE   68                                      ;
    0000        00706           .WORD   0                                       ;
00000000        00708           .LONG   0                                       ;
00000000        0070C           .LONG   0                                       ;
00000000        00710           .LONG   0                                       ;
    0000#       00714           .WORD   0[3]                                    ;
    0000        0071A           .WORD   0                                       ;
00000000        0071C           .LONG   0                                       ;
    0000        00720           .WORD   0                                       ;
      00        00722           .BYTE   0                                       ;
      00        00723           .BYTE   0                                       ;
    0584        00724           .WORD   1412                                    ;
    0000        00726           .WORD   0                                       ;
00000000        00728           .LONG   0                                       ;
00000000        0072C           .LONG   0                                       ;
00000000        00730           .LONG   0                                       ;
00000000'       00734           .ADDRESS RECBUF+4                               ;
      20        00738           .BYTE   32                                      ;
      00        00739           .BYTE   0                                       ;
      00        0073A           .BYTE   0                                       ;
      00        0073B           .BYTE   0                                       ;
00000000        0073C           .LONG   0                                       ;
00000000'       00740           .ADDRESS UAFFAB                                 ;
00000000        00744           .LONG   0                                       ;
```

```
      01  00748 NAFRAB::.BYTE    1
      44  00749          .BYTE    68
    0000  0074A          .WORD    0
00000000  0074C          .LONG    0
00000000  00750          .LONG    0
00000000  00754          .LONG    0
    0000# 00758          .WORD    0[3]
    0000  0075E          .WORD    0
00000000  00760          .LONG    0
    0000  00764          .WORD    0
      01  00766          .BYTE    1
      00  00767          .BYTE    0
    0064  00768          .WORD    100
    0064  0076A          .WORD    100
00000000  0076C          .LONG    0
00000000  00770          .LONG    0
00000000  00774          .LONG    0
00000000' 00778          .ADDRESS NETBUF
      40  0077C          .BYTE    64
      00  0077D          .BYTE    0
      00  0077E          .BYTE    0
      00  0077F          .BYTE    0
00000000  00780          .LONG    0
00000000' 00784          .ADDRESS NAFFAB
00000000  00788          .LONG    0
          0078C TIME_BUF::
                         .BLKB    8
          00794 PWD_FLAG::
                         .BLKB    4
      01  00798 OUTRAB::.BYTE    1
      44  00799          .BYTE    68
    0000  0079A          .WORD    0
00000000  0079C          .LONG    0
00000000  007A0          .LONG    0
00000000  007A4          .LONG    0
    0000# 007A8          .WORD    0[3]
    0000  007AE          .WORD    0
00000000  007B0          .LONG    0
    0000  007B4          .WORD    0
      00  007B6          .BYTE    0
      00  007B7          .BYTE    0
    0000  007B8          .WORD    0
    0000  007BA          .WORD    0
00000000  007BC          .LONG    0
00000000' 007C0          .ADDRESS DISBUF
00000000  007C4          .LONG    0
00000000  007C8          .LONG    0
      00  007CC          .BYTE    0
      00  007CD          .BYTE    0
      00  007CE          .BYTE    0
      00  007CF          .BYTE    0
00000000  007D0          .LONG    0
00000000' 007D4          .ADDRESS OUTFAB
00000000  007D8          .LONG    0
          007DC FOUND_MATCH::
                         .BLKB    4
          007E0 UIC_   G::
```

```
                        .BLKB   4
007E4 GRP_WILD::
                        .BLKB   4
007E8 MEM_WILD::
                        .BLKB   4
007EC STR_WILD::
                        .BLKB   4

SD_TOKEN1=              P.AAA
SD_TOKEN2=              P.AAC
SD_BRIEF=               P.AAE
SD_FULL=                P.AAG
SD_ADD_IDENTIFIER=      P.AAI
SD_REMOVE_IDENTIFIER=
                        P.AAK
SD_MODIFY_IDENTIFIER=
                        P.AAM
ENCRYPT==               2
DISBUFLEN==             132
SYSUAF_STRING=          P.AAP
NETUAF_STRING=          P.AAQ
MOD_ACT_DSC=            P.AAR
ADD_ACT_DSC=            P.AAT
REM_ACT_DSC=            P.AAV
FAO_LIN_DSC=            P.AAX
DBL_COLON=              P.AAZ
WAKEDELTA=              P.ABB
DEFUSER=                P.ABC
DEFPASS=                P.ABD
DEFCLITABL=             P.ABE
DEFACT=                 P.ABF
DEFCLI=                 P.ABG
DEFOWNER=               P.ABH
DEFLGICMD=              P.ABI
DEFGRP=                 128
DEFMEM=                 128
DEFDIR=                 P.ABJ
DEFDEV=                 P.ABK
DEFBIOLM=               6
DEFBYTLM=               4096
DEFDIOLM=               6
DEFFILLM=               20
DEFFLAGS=               0
DEFTQCNT=               10
DEFPRCCNT=              2
DEFPRI=                 4
DEFQUEPRI=              4
DEFWSQUOTA=             200
DEFDFWSCNT=             150
DEFWSEXTENT=            500
DEFCPUTIM=              0
DEFASTLM=               10
DEFPGFLQUOTA=           10000
DEFENQLM=               10
DEFPBYTLM=              0
DEFSHRFILLM=            0
DEFMAXJOBS=             0
```

```
DEFMAXACCTJOBS=     0
DEFMAXDETACH=       0
DEFJTQUOTA=         1024
DEFPRIMEDAYS=       96
DEFHOURS=           0
DEFPRIV=            F.ABL
DEFPWDLENGTH=       6
DEFPWDLIFE=         P.ABM
SYSUSER=            P.ABN
SYSPASS=            P.ABO
SYSCLITABL=         P.ABP
SYSACT=             P.ABQ
SYSCLI=             P.ABR
SYSOWNER=           P.ABS
SYSLGICMD=          P.ABT
SYSGRP=             1
SYSMEM=             4
SYSDIR=             P.ABU
SYSDEV=             P.ABV
SYSBIOLM=           12
SYSBYTLM=           20480
SYSDIOLM=           12
SYSFILLM=           20
SYSFLAGS=           0
SYSTQCNT=           20
SYSPRCCNT=          10
SYSPRI=             4
SYSQUEPRI=          4
SYSWSQUOTA=         350
SYSWSEXTENT=        1024
SYSDFWSCNT=         150
SYSCPUTIM=          0
SYSASTLM=           20
SYSPGFLQUOTA=       10000
SYSENQLM=           20
SYSPBYTLM=          0
SYSSHRFILLM=        0
SYSMAXJOBS=         0
SYSMAXDETACH=       0
SYSJTQUOTA=         1024
SYSMAXACCTJOBS=     0
SYSPRIMEDAYS=       96
SYSHOURS=           0
SYSPRIV=            P.ABW
SYSPWDLENGTH=       8
SYSPWDLIFE=         P.ABX
TOKENLEN==          TOKENDSC
TOKENPTR==          TOKENDSC+4
REC_USER_DSC==      P.ACG
REC_ENCRYPT_DSC==   P.ACH
SYMBOL_STR==        P.ACI
ACCPRMPT=           P.ACJ
ACCPRMPT2=          P.ACK
NEWMSG20=           P.ACL
        .EXTRN  CLI$_BUFOVF, CLI$_NOCLINT
        .EXTRN  LBR$OUTPUT_HELP
        .EXTRN  LIB$GET_FOREIGN
```

```
                .EXTRN  LIB$GET_INPUT, LIB$PUT_OUTPUT
                .EXTRN  FMG$MATCH_NAME, CLI$DCL_PARSE
                .EXTRN  CLI$DISPATCH, CLI$PRESENT
                .EXTRN  CLI$GET_VALUE, UPDATE_RECORD
                .EXTRN  PARSE_WILD, LGI$HPWD
                .EXTRN  UAF$ADD_IDENT_RECBUF
                .EXTRN  UAF$BUILD_HOLDER
                .EXTRN  UAF$FIND_UIC, UAF$REMOVE_IDENT_RECBUF
                .EXTRN  UAF$WRITE_RIGHTS
                .EXTRN  EXE$GL_SYSUIC, RDB_HEADER_FLAG
                .EXTRN  RDB_LIST_FLAG, ATTRIBUTES
                .EXTRN  HOLDER, IDENT, AUTHORIZE_COMMANDS
                .EXTRN  PRV$AB_NAMES, LIB$SIGNAL
                .EXTRN  UAF$_ADDERR, UAF$_ADDMSG
                .EXTRN  UAF$_BADNODFORM
                .EXTRN  UAF$_BADSPC, UAF$_BADUSR
                .EXTRN  UAF$_CLIWARNMSG
                .EXTRN  UAF$_CMDTOOLONG
                .EXTRN  UAF$_CONERR, UAF$_COPMSG
                .EXTRN  UAF$_CREERR, UAF$_DEFERR
                .EXTRN  UAF$_DEFPWD, UAF$_DONEMSG
                .EXTRN  UAF$_GETERR, UAF$_HELPERR
                .EXTRN  UAF$_INVCMD, UAF$_INVRSP
                .EXTRN  UAF$_INVUSERNAME
                .EXTRN  UAF$_KEYNOTFND, UAF$_KEYNOTUNQ
                .EXTRN  UAF$_LSTERR, UAF$_LSTMSG1
                .EXTRN  UAF$_LSTMSG2, UAF$_MDFYERR
                .EXTRN  UAF$_MDFYMSG, UAF$_NAFADDERR
                .EXTRN  UAF$_NAFADDMSG, UAF$_NAFAEX
                .EXTRN  UAF$_NAFCONERR, UAF$_NAFCREERR
                .EXTRN  UAF$_NAFDNE, UAF$_NAFDONEMSG
                .EXTRN  UAF$_NAFNOMODS, UAF$_NAFUAEERR
                .EXTRN  UAF$_NAMETOOBIG
                .EXTRN  UAF$_NETLSTMSG, UAF$_NAOFIL
                .EXTRN  UAF$_NAONAF, UAF$_NOARG
                .EXTRN  UAF$_NODEFPWD, UAF$_NODTOOBIG
                .EXTRN  UAF$_NOMODS, UAF$_NOTUNQ
                .EXTRN  UAF$_NOUSERNAME
                .EXTRN  UAF$_PREMMSG, UAF$_PUTERR
                .EXTRN  UAF$_RDBDONEMSG
                .EXTRN  UAF$_RDBMDFYERR
                .EXTRN  UAF$_RDBMDFYERRU
                .EXTRN  UAF$_RDBMDFYMSG
                .EXTRN  UAF$_RDBNOMODS, UAF$_REMDEF
                .EXTRN  UAF$_REMERR, UAF$_REMMSG
                .EXTRN  UAF$_REMSYS, UAF$_RENDEF
                .EXTRN  UAF$_RENMSG, UAF$_RENSYS
                .EXTRN  UAF$_RONLY, UAF$_SHOW_ERR
                .EXTRN  UAF$_SYSMSG1, UAF$_SYSMSG2
                .EXTRN  UAF$_UAEERR, UAF$_UICERR
                .EXTRN  UAF$_ZISQUAL, UAF$_ZZPRACREN
                .EXTRN  SYS$OPEN, SYS$CONNECT
                .EXTRN  SYS$CREATE

                .PSECT  $CODE$,NOWRT,2

0FFC 00000 START:  .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11          ; 0903
```

```
                        5B 00000000G   00  9E 00002          MOVAB   LIB$GET_INPUT, R11                      ;
                        5A 00000000G   00  9E 00009          MOVAB   SYS$CONNECT, R10                        ;
                        59 00000000G   8F  D0 00010          MOVL    #CLI$_NOCLINT, R9                       ;
                        58 00000000G   00  9E 00017          MOVAB   LIB$STOP, R8                            ;
                        57 00000000'   00  9E 0001E          MOVAB   CMDLINDSC, R7                           ;
                        56 00000000'   00  9E 00025          MOVAB   STATUS, R6                              ;
                              14       A7  94 0002C          CLRB    RIGHTSLIST_MODIFIED                     ; 0942
                            EFC4       C6  7C 0002F          CLRQ    MODIFY_FLAG                             ; 0944
                            F9B8       C6  9F 00033          PUSHAB  INFAB                                   ; 0950
          00000000G     00             01  FB 00037          CALLS   #1, SYS$OPEN                            ;
                        66             50  D0 0003E          MOVL    R0, STATUS                              ;
                  10    A7             50  D0 00041          MOVL    R0, RMSERR                              ;
                        05             50  E8 00045          BLBS    R0, 1$                                  ;
                                       66  DD 00048          PUSHL   STATUS                                  ; 0952
                        68             01  FB 0004A          CALLS   #1, LIB$STOP                            ;
                            FA08       C6  9F 0004D 1$:      PUSHAB  INRAB                                   ; 0954
                        6A             01  FB 00051          CALLS   #1, SYS$CONNECT                         ;
                        66             50  D0 00054          MOVL    R0, STATUS                              ;
                  10    A7             50  D0 00057          MOVL    R0, RMSERR                              ;
                        05             50  E8 0005B          BLBS    R0, 2$                                  ;
                                       66  DD 0005E          PUSHL   STATUS                                  ; 0956
                        68             01  FB 00060          CALLS   #1, LIB$STOP                            ;
                            FA4C       C6  9F 00063 2$:      PUSHAB  OUTFAB                                  ; 0958
          00000000G     00             01  FB 00067          CALLS   #1, SYS$CREATE                          ;
                            0694       C7  9F 0006E          PUSHAB  OUTRAB                                  ; 0959
                        6A             01  FB 00072          CALLS   #1, SYS$CONNECT                         ;
          00000000V     00             00  FB 00075          CALLS   #0, SETUP                               ; 0961
                                       57  DD 0007C          PUSHL   R7                                      ; 0968
          00000000G     00             01  FB 0007E          CALLS   #1, LIB$GET_FOREIGN                     ;
                        38             50  E9 00085          BLBC    R0, 4$                                  ;
                                       67  B5 00088          TSTW    CMDLINDSC                               ;
                                       34  13 0008A          BEQL    4$                                      ;
                              00000000'    9F 0008C          PUSHAB  P.ACM                                   ; 0975
                                       5B  DD 00092          PUSHL   R11                                     ;
                                       7E  D4 00094          CLRL    -(SP)                                   ;
                              00000000G    9F 00096          PUSHAB  AUTHORIZE_COMMANDS                      ;
                                       57  DD 0009C          PUSHL   R7                                      ;
          00000000G     00             05  FB 0009E          CALLS   #5, CLI$DCL_PARSE                       ;
                        66             50  D0 000A5          MOVL    R0, STATUS                              ;
                        09             50  E9 000A8          BLBC    R0, 3$                                  ;
          00000000G     00             00  FB 000AB          CALLS   #0, CLI$DISPATCH                        ; 0978
                                       51  11 000B2          BRB     7$                                      ; 0979
                        59             66  D1 000B4 3$:      CMPL    STATUS, R9                              ; 0985
                                       45  13 000B7          BEQL    6$                                      ;
          00000000V     00             00  FB 000B9          CALLS   #0, ACC$EXIT                            ; 0980
          00000000V     00             00  FB 000C0 4$:      CALLS   #0, GET_CMD_LINE                        ; 1002
                        F6             50  E9 000C7          BLBC    R0, 4$                                  ;
                              00000000'    9F 000CA          PUSHAB  P.ACO                                   ; 1004
                                       5B  DD 000D0          PUSHL   R11                                     ;
                                       7E  D4 000D2          CLRL    -(SP)                                   ;
                              00000000G    9F 000D4          PUSHAB  AUTHORIZE_COMMANDS                      ;
                                       57  DD 000DA          PUSHL   R7                                      ;
          00000000G     00             05  FB 000DC          CALLS   #5, CLI$DCL_PARSE                       ;
                        66             50  D0 000E3          MOVL    R0, STATUS                              ;
                        10             50  E9 000E6          BLBC    R0, 5$                                  ;
    08          00        6E             00  2C 000E9          MOVC5   #0, (SP), #0, #8, UAF$GL_CTLMSK         ; 1007
                                 9C    A7     000EE                                                         ;
```

```
                00000000G  00        00  FB  000F0          CALLS   #0, CLI$DISPATCH                 ; 1008
                                      C7  11  000F7          BRB     4$                              ; 1004
                           59         66  D1  000F9  5$:     CMPL    STATUS, R9                      ; 1011
                                      C2  12  000FC          BNEQ    4$                              ;
                                      59  DD  000FE  6$:     PUSHL   R9                              ; 1013
                           68         01  FB  00100          CALLS   #1, LIB$STOP                    ;
                                      BB  11  00103          BRB     4$                              ; 1004
                           50         01  D0  00105  7$:     MOVL    #1, R0                          ; 1016
                                          04  00108          RET                                     ; 1018

; Routine Size:  265 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  933          1019  1 %sbttl 'setup - open initial files'
;  934          1020  1 routine setup : novalue =
;  935          1021  2 begin
;  936          1022  2
;  937          1023  2 !++
;  938          1024  2 !
;  939          1025  2 ! FUNCTIONAL DESCRIPTION:
;  940          1026  2 !
;  941          1027  2 !       This routine does all of the initial file manipulation for the
;  942          1028  2 !       program. It determines whether or not a previous SYSUAF.DAT exists.
;  943          1029  2 !       If not, it creates one (if the user wishes to proceed).
;  944          1030  2 !
;  945          1031  2 ! INPUTS:
;  946          1032  2 !
;  947          1033  2 !       none
;  948          1034  2 !
;  949          1035  2 ! IMPLICIT INPUTS:
;  950          1036  2 !
;  951          1037  2 !       none
;  952          1038  2 !
;  953          1039  2 ! OUTPUTS:
;  954          1040  2 !
;  955          1041  2 !       none
;  956          1042  2 !
;  957          1043  2 ! IMPLICIT OUTPUTS:
;  958          1044  2 !
;  959          1045  2 !       none
;  960          1046  2 !
;  961          1047  2 ! ROUTINE VALUE:
;  962          1048  2 !
;  963          1049  2 !       none
;  964          1050  2 !
;  965          1051  2 ! SIDE EFFECTS:
;  966          1052  2 !
;  967          1053  2 !       none
;  968          1054  2 !--
;  969          1055  2
;  970          1056  2 local
;  971          1057  2         status          : long,
;  972          1058  2         curpriv         : vector [2],
;  973          1059  2         procpriv        : vector [2],
;  974          1060  2         item_list       : block [64, byte],
;  975          1061  2         newfile;                        ! indicates new file must
;  976          1062  2                                         ! be created.
;  977          1063  2
;  978          1064  2 item_list [0,0,16,0] = 4;
;  979          1065  2 item_list [2,0,16,0] = jpi$_pid;
;  980          1066  2 item_list [4,0,32,0] = pid;
;  981          1067  2 item_list [8,0,32,0] = 0;
;  982          1068  2 item_list [12,0,16,0] = 12;
;  983          1069  2 item_list [14,0,16,0] = jpi$_username;
;  984          1070  2 item_list [16,0,32,0] = username_buf;
;  985          1071  2 item_list [20,0,32,0] = username_dsc[0];
;  986          1072  2 item_list [24,0,16,0] = 8;
;  987          1073  2 item_list [26,0,16,0] = jpi$_curpriv;
;  988          1074  2 item_list [28,0,32,0] = curpriv;
;  989          1075  2 item_list [32,0,32,0] = 0;
```

```
;  990     1076  2 item_list [36,0,16,0] = 8;
;  991     1077  2 item_list [38,0,16,0] = jpi$_procpriv;
;  992     1078  2 item_list [40,0,32,0] = procpriv;
;  993     1079  2 item_list [44,0,32,0] = 0;
;  994     1080  2 item_list [48,0,16,0] = 4;
;  995     1081  2 item_list [50,0,16,0] = jpi$_sts;
;  996     1082  2 item_list [52,0,32,0] = pcb_sts;
;  997     1083  2 item_list [56,0,32,0] = 0;
;  998     1084  2 item_list [60,0,32,0] = 0;
;  999     1085  2
; 1000     1086  2 $getjpi (itmlst = item_list);          ! Obtain pid, username and privs
; 1001     1087  2
; 1002     1088  2 username_dsc[0] = namelen (uaf$s_username, username_buf);
; 1003     1089  2
; 1004     1090  2 !**************************************************************************
; 1005     1091  2 !
; 1006     1092  2 !                          Open SYSUAF.DAT
; 1007     1093  2 !
; 1008     1094  2 !**************************************************************************
; 1009     1095  2
; 1010     1096  2 newfile = false;                       ! note no new file yet
; 1011     1097  2
; 1012     1098  2 if rmsbad ($open (fab = uaffab))
; 1013     1099  2 then
; 1014     1100  2 !
; 1015     1101  2 ! sysuaf.dat doesn't exist
; 1016     1102  2 !
; 1017     1103  3     begin
; 1018     1104  3     LIB$SIGNAL(UAF$_NAOFIL, 0, .rmserr);
; 1019     1105  3     if .rmserr eql rms$_fnf
; 1020     1106  3     then
; 1021     1107  4         begin
; 1022     1108  4         while true
; 1023     1109  4         do
; 1024     1110  4 !
; 1025     1111  4 ! Ask if a new one is desired
; 1026     1112  4 !
; 1027     1113  5                 begin
; 1028     1114  5                 ask (newmsg20, cmdbuf[0], cmdbuflen);
; 1029     1115  5                 if .cmdbuf [0] eql 'Y'
; 1030     1116  5                 then exitloop newfile = true;
; 1031     1117  5                 if .cmdbuf [0] eql 'N'
; 1032     1118  5                 then acc$exit ();
; 1033     1119  5                 LIB$SIGNAL(UAF$_INVRSP);
; 1034     1120  4                 end;
; 1035     1121  4         end
; 1036     1122  3     else
; 1037     1123  3         acc$exit ();
; 1038     1124  2     end;
; 1039     1125  2
; 1040     1126  2
; 1041     1127  2 !
; 1042     1128  2 ! The file will be created if it does not already exist.
; 1043     1129  2 ! In any case connect the RAB.
; 1044     1130  2 !
; 1045     1131  2
; 1046     1132  2 if .newfile
```

```
;  1047     1133  2 then
;  1048     1134  2 !
;  1049     1135  2 ! A new file is requested
;  1050     1136  2 !
;  1051     1137  2     if rmsbad ($create (fab = uaffab))
;  1052     1138  2     then
;  1053     1139  2 !
;  1054     1140  2 ! Quit regardless of error on a $CREATE: don't
;  1055     1141  2 ! want to give read-only user ability to create a file
;  1056     1142  2 !
;  1057     1143  2         LIB$SIGNAL(UAF$_CREERR, 0, .rmserr);
;  1058     1144  2
;  1059     1145  2 if rmsbad ($connect (rab = uafrab))
;  1060     1146  2 then LIB$SIGNAL(UAF$_CONERR, 0, .rmserr);
;  1061     1147  2 uafrab[rab$b_rac] = rab$c_key;          ! normal access is by key
;  1062     1148  2
;  1063     1149  2 !
;  1064     1150  2 ! Check to see if there was no old file to use.  If so write a default and
;  1065     1151  2 ! a system manager record.
;  1066     1152  2 !
;  1067     1153  2
;  1068     1154  2 if .newfile
;  1069     1155  2 then
;  1070     1156  3     begin
;  1071     1157  3     modify_flag = true;                 ! must rename when done
;  1072     1158  3     build_ini_recs ();                  ! build default and system manager records
;  1073     1159  3     uafrab[rab$w_rsz] = uaf$c_fixed;
;  1074     1160  3
;  1075     1161  3     default_size = uaf$c_fixed;
;  1076     1162  3     uafrab[rab$l_rbf] = default_record; ! insert default record address
;  1077     1163  4     if rmsbad ($put (rab = uafrab))
;  1078     1164  4     then LIB$SIGNAL(UAF$_PUTERR, 0, .rmserr);    ! report error
;  1079     1165  4     uafrab[rab$l_rbf] = recbuf;                  ! establish proper address (and
;  1080     1166  4                                                  ! address of system record)
;  1081     1167  4     if rmsbad ($put (rab = uafrab))              ! output system record
;  1082     1168  4     then LIB$SIGNAL(UAF$_PUTERR, 0, .rmserr);
;  1083     1169  3     end
;  1084     1170  2 else
;  1085     1171  2 !
;  1086     1172  2 ! Read in the default record.
;  1087     1173  2 !
;  1088     1174  3     begin
;  1089     1175  3     uafrab[rab$l_ubf] = default_record;
;  1090     1176  3     if not locate_user (.defuser<0,8>, defuser+1, false)
;  1091     1177  3     then LIB$SIGNAL(UAF$_DEFERR, 0, .rmserr);
;  1092     1178  3     default_size = .uafrab[rab$w_rsz];
;  1093     1179  3     end;
;  1094     1180  2
;  1095     1181  2 uafrab[rab$l_ubf] = recbuf;               ! establish proper addresses
;  1096     1182  2 uafrab[rab$l_rbf] = recbuf;
;  1097     1183  2
;  1098     1184  2 !**************************************************************************
;  1099     1185  2 !
;  1100     1186  2 !                           Open NETUAF.DAT
;  1101     1187  2 !
;  1102     1188  2 !**************************************************************************
;  1103     1189  2
```

```
;  1104          1190  2 netuaf_exists = true;                    ! Assume NETUAF.DAT exists...
;  1105          1191  2
;  1106          1192  2 !
;  1107          1193  2 ! Try to open NETUAF.DAT and see what happens...
;  1108          1194  2 !
;  1109          1195  2 if rmsbad ($open (fab = naffab))
;  1110          1196  2 then
;  1111          1197  2 !
;  1112          1198  2 ! Couldn't open it
;  1113          1199  2 !
;  1114          1200  2     if .rmserr eql rms$_fnf
;  1115          1201  2     then
;  1116          1202  2         netuaf_exists = false          ! it doesn't exist
;  1117          1203  2     else
;  1118          1204  2         LIB$SIGNAL(UAF$_NAONAF, 0, .rmserr)     ! open error for some other reason
;  1119          1205  2
;  1120          1206  2 else
;  1121          1207  2 !
;  1122          1208  2 ! NETUAF.DAT opened without error
;  1123          1209  2 !
;  1124          1210  2     if rmsbad ($connect (rab = nafrab))
;  1125          1211  2     then LIB$SIGNAL(UAF$_NAFCONERR)             ! connect error
;  1126          1212  2 !
;  1127          1213  2 ! Everything opened and connected, establish proper NETUAF addresses
;  1128          1214  2 !
;  1129          1215  2     else
;  1130          1216  3         begin
;  1131          1217  3         nafrab [rab$l_ubf] = netbuf;
;  1132          1218  3         nafrab [rab$l_rbf] = netbuf;
;  1133          1219  2         end;
;  1134          1220  2
;  1135          1221  2 !
;  1136          1222  2 ! Check to see if the rights data base exists.  Try to translate an ID
;  1137          1223  2 ! and if we get a file not found error then it doesn't exist.
;  1138          1224  2 !
;  1139          1225  2 ident[uic$v_format] = uic$k_uic_format ;
;  1140          1226  2 ident[uic$v_group]  = 1 ;
;  1141          1227  2 ident[uic$v_member] = 4 ;
;  1142          1228  2 status = $idtoasc ( id = .ident ) ;
;  1143          1229  2 if .status eql rms$_fnf
;  1144          1230  2     then rdb_exists = false
;  1145          1231  2     else rdb_exists = true ;
;  1146          1232  2
;  1147          1233  1 end;


                                                .EXTRN  SYS$GETJPI, SYS$PUT
                                                .EXTRN  SYS$IDTOASC

                                 0FFC 00000 SETUP:  .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11          ; 1020
                5B 00000000G   00  9E 00002        MOVAB   SYS$PUT, R11                                    ;
                5A 00000000G   00  9E 00009        MOVAB   SYS$CONNECT, R10                                ;
                59 00000000V   00  9E 00010        MOVAB   ACC$EXIT, R9                                    ;
                58 00000000G   00  9E 00017        MOVAB   SYS$OPEN, R8                                    ;
                57 00000000'   00  9E 0001E        MOVAB   NEWMSG20, R7                                    ;
                56 00000000G   00  9E 00025        MOVAB   IDENT, R6                                       ;
```

```
                                 55 00000000G   00   9E 0002C         MOVAB   LIB$SIGNAL, R5                                  ;
                                 54 00000000'   00   9E 00033         MOVAB   USERNAME_BUF, R4                                ;
                                 53 00000000'   00   9E 0003A         MOVAB   RMSERR, R3                                      ;
                                 5E        B4   AE   9E 00041         MOVAB   -76(SP), SP                                     ;
                                    03190004   8F   DD 00045         PUSHL   #51970052                                       ;                1064
                              04    AE         24   A4   9E 0004B     MOVAB   PID, ITEM_LIST+4                                ;                1066
                                          08   AE   D4 00050         CLRL    ITEM_LIST+8                                     ;                1067
                              0C    AE 0202000C   8F   D0 00053     MOVL    #33685516, ITEM_LIST+12                         ;                1068
                              10    AE         64   9E 0005B         MOVAB   USERNAME_BUF, ITEM_LIST+16                      ;                1070
                              14    AE         28   A4   9E 0005F     MOVAB   USERNAME_DSC, ITEM_LIST+20                      ;                1071
                              18    AE 04000008   8F   D0 00064     MOVL    #67108872, ITEM_LIST+24                         ;                1072
                              1C    AE         48   AE   9E 0006C     MOVAB   CURPRIV, ITEM_LIST+28                           ;                1074
                                          20   AE   D4 00071         CLRL    ITEM_LIST+32                                    ;                1075
                              24    AE 02040008   8F   D0 00074     MOVL    #33816584, ITEM_LIST+36                         ;                1076
                              28    AE         40   AE   9E 0007C     MOVAB   PROCPRIV, ITEM_LIST+40                          ;                1078
                                          2C   AE   D4 00081         CLRL    ITEM_LIST+44                                    ;                1079
                              30    AE 03050004   8F   D0 00084     MOVL    #50659332, ITEM_LIST+48                         ;                1080
                              34    AE         20   A4   9E 0008C     MOVAB   PCB_STS, ITEM_LIST+52                           ;                1082
                                             38   AE   7C 00091         CLRQ    ITEM_LIST+56                                    ;                1083
                                                  7E   7C 00094         CLRQ    -(SP)                                           ;                1086
                                                  7E   D4 00096         CLRL    -(SP)                                           ;
                                             0C   AE   9F 00098         PUSHAB  ITEM_LIST                                       ;
                                                  7E   7C 0009B         CLRQ    -(SP)                                           ;
                                                  7E   D4 0009D         CLRL    -(SP)                                           ;
                           00000000G  00            07   FB 0009F     CALLS   #7, SYS$GETJPI                                  ;
                        64            20            20   3A 000A6     LOCC    #32, #32, USERNAME_BUF                          ;                1088
                     28 A4            20            50   C3 000AA     SUBL3   R0, #32, USERNAME_DSC                           ;
                                                    52   D4 000AF     CLRL    NEWFILE                                         ;                1096
                                        0EF0        C4   9F 000B1     PUSHAB  UAFFAB                                          ;                1098
                                    68              01   FB 000B5     CALLS   #1, SYS$OPEN                                    ;
                                    63              50   D0 000B8     MOVL    R0, RMSERR                                      ;
                                    4F              50   E8 000BB     BLBS    R0, 5$                                          ;
                                                    63   DD 000BE     PUSHL   RMSERR                                          ;                1104
                                                    7E   D4 000C0     CLRL    -(SP)                                           ;
                                 00000000G          8F   DD 000C2     PUSHL   #UAF$_NAOFIL                                    ;
                                    65              03   FB 000C8     CALLS   #3, LIB$SIGNAL                                  ;
                     00018292       8F              63   D1 000CB     CMPL    RMSERR, #98962                                  ;                1105
                                                    36   12 000D2     BNEQ    4$                                              ;
                                 7E        0400      8F   3C 000D4 1$: MOVZWL  #1024, -(SP)                                   ;                1114
                                             0098   C4   9F 000D9     PUSHAB  CMDBUF                                          ;
                                                    57   DD 000DD     PUSHL   R7                                              ;
                           00000000V  00            03   FB 000DF     CALLS   #3, ASK                                         ;
                                    50              0098 C4   9A 000E6 MOVZBL  CMDBUF, R0                                      ;                1115
                                 59 8F              50   91 000EB     CMPB    R0, #89                                         ;
                                                    05   12 000EF     BNEQ    2$                                              ;
                                    52              01   D0 000F1     MOVL    #1, NEWFILE                                     ;                1116
                                                    17   11 000F4     BRB     5$                                              ;
                                 4E 8F              50   91 000F6 2$: CMPB    R0, #78                                         ;                1117
                                                    03   12 000FA     BNEQ    3$                                              ;
                                    69              00   FB 000FC     CALLS   #0, ACC$EXIT                                    ;                1118
                                 00000000G          8F   DD 000FF 3$: PUSHL   #UAF$_INVRSP                                    ;                1119
                                    65              01   FB 00105     CALLS   #1, LIB$SIGNAL                                  ;
                                                    CA   11 00108     BRB     1$                                              ;                1108
                                    69              00   FB 0010A 4$: CALLS   #0, ACC$EXIT                                    ;                1123
                                    1E              52   E9 0010D 5$: BLBC    NEWFILE, 6$                                     ;                1132
                                             0EF0   C4   9F 00110     PUSHAB  UAFFAB                                          ;                1137
                           00000000G  00            01   FB 00114     CALLS   #1, SYS$CREATE                                  ;
```

```
                63              50  D0 0011B          MOVL    R0, RMSERR                      ;
                0D              50  E8 0011E          BLBS    R0, 6$                          ;
                                63  DD 00121          PUSHL   RMSERR                          ; 1143
                                7E  D4 00123          CLRL    -(SP)                           ;
                00000000G       8F  DD 00125          PUSHL   #UAF$_CREERR                    ;
                65              03  FB 00128          CALLS   #3, LIB$SIGNAL                  ;
                    05F0        C3  9F 0012E 6$:      PUSHAB  UAFRAB                          ; 1145
                6A              01  FB 00132          CALLS   #1, SYS$CONNECT                 ;
                63              50  D0 00135          MOVL    R0, RMSERR                      ;
                0D              50  E8 00138          BLBS    R0, 7$                          ;
                                63  DD 0013B          PUSHL   RMSERR                          ; 1146
                                7E  D4 0013D          CLRL    -(SP)                           ;
                00000000G       8F  DD 0013F          PUSHL   #UAF$_CONERR                    ;
                65              03  FB 00145          CALLS   #3, LIB$SIGNAL                  ;
            060E C3             01  90 00148 7$:      MOVB    #1, UAFRAB+30                   ; 1147
                5C              52  E9 0014D          BLBC    NEWFILE, 9$                     ; 1154
            44  A4              01  D0 00150          MOVL    #1, MODIFY_FLAG                 ; 1157
00000000V   00                  00  FB 00154          CALLS   #0, BUILD_INI_RECS              ; 1158
          0612  C3      0284    8F  B0 00158          MOVW    #644, UAFRAB+34                 ; 1159
          0498  C4      0284    8F  B0 00162          MOVW    #644, DEFAULT_SIZE              ; 1161
          0618  C3      049C    C4  9E 00169          MOVAB   DEFAULT_RECORD, UAFRAB+40       ; 1162
                05F0            C3  9F 00170          PUSHAB  UAFRAB                          ; 1163
                6B              01  FB 00174          CALLS   #1, SYS$PUT                     ;
                63              50  D0 00177          MOVL    R0, RMSERR                      ;
                0D              50  E8 0017A          BLBS    R0, 8$                          ;
                                63  DD 0017D          PUSHL   RMSERR                          ; 1164
                                7E  D4 0017F          CLRL    -(SP)                           ;
                00000000G       8F  DD 00181          PUSHL   #UAF$_PUTERR                    ;
                65              03  FB 00187          CALLS   #3, LIB$SIGNAL                  ;
            0618 C3       08    A3  9E 0018A 8$:      MOVAB   RECBUF, UAFRAB+40               ; 1165
                05F0            C3  9F 00190          PUSHAB  UAFRAB                          ; 1167
                6B              01  FB 00194          CALLS   #1, SYS$PUT                     ;
                63              50  D0 00197          MOVL    R0, RMSERR                      ;
                3F              50  E8 0019A          BLBS    R0, 11$                         ;
                                63  DD 0019D          PUSHL   RMSERR                          ; 1168
                                7E  D4 0019F          CLRL    -(SP)                           ;
                00000000G       8F  DD 001A1          PUSHL   #UAF$_PUTERR                    ;
                65              03  FB 001A7          CALLS   #3, LIB$SIGNAL                  ;
                                30  11 001AA          BRB     11$                             ; 1154
            0614 C3       049C    C4  9E 001AC 9$:      MOVAB   DEFAULT_RECORD, UAFRAB+36       ; 1175
                                7E  D4 001B3          CLRL    -(SP)                           ; 1176
                    FEFB        C7  9F 001B5          PUSHAB  DEFUSER+1                       ;
                7E  FEFA        C7  9A 001B9          MOVZBL  DEFUSER, -(SP)                  ;
00000000V       00              03  FB 001BE          CALLS   #3, LOCATE_USER                 ;
                0D              50  E8 001C5          BLBS    R0, 10$                         ;
                                63  DD 001C8          PUSHL   RMSERR                          ; 1177
                                7E  D4 001CA          CLRL    -(SP)                           ;
                00000000G       8F  DD 001CC          PUSHL   #UAF$_DEFERR                    ;
                65              03  FB 001D2          CALLS   #3, LIB$SIGNAL                  ;
          0498  C4      0612    C3  B0 001D5 10$:     MOVW    UAFRAB+34, DEFAULT_SIZE         ; 1178
          0614  C3        08    A3  9E 001DC 11$:     MOVAB   RECBUF, UAFRAB+36               ; 1181
          0618  C3        08    A3  9E 001E2          MOVAB   RECBUF, UAFRAB+40               ; 1182
            F8  A3              01  D0 001E8          MOVL    #1, NETUAF_EXISTS               ; 1190
                1030            C4  9F 001EC          PUSHAB  NAFFAB                          ; 1195
                68              01  FB 001F0          CALLS   #1, SYS$OPEN                    ;
                63              50  D0 001F3          MOVL    R0, RMSERR                      ;
                20              50  E8 001F6          BLBS    R0, 13$                         ;
```

```
                                50                  63  D0 001F9          MOVL    RMSERR, R0                          ; 1200
                      00018292  8F                  50  D1 001FC          CMPL    R0, #98962
                                                    05  12 00203          BNEQ    12$
                                         F8         A3  D4 00205          CLRL    NETUAF_EXISTS                       ; 1202
                                                    35  11 00208          BRB     15$
                                                    50  DD 0020A 12$:     PUSHL   R0                                  ; 1204
                                                    7E  D4 0020C          CLRL    -(SP)
                                00000000G           8F  DD 0020E          PUSHL   #UAF$_NAONAF
                                65                      03  FB 00214      CALLS   #3, LIB$SIGNAL
                                                        26  11 00217      BRB     15$
                                          0634          C3  9F 00219 13$: PUSHAB  NAFRAB                              ; 1200
                                6A                      01  FB 0021D      CALLS   #1, SYS$CONNECT                     ; 1210
                                63                      50  D0 00220      MOVL    R0, RMSERR
                                0B                      50  E8 00223      BLBS    R0, 14$
                                00000000G               8F  DD 00226      PUSHL   #UAF$_NAFCONERR                     ; 1211
                                65                      01  FB 0022C      CALLS   #1, LIB$SIGNAL
                                                        0E  11 0022F      BRB     15$
                      0658      C3          058C      C3  9E 00231 14$:     MOVAB   NETBUF, NAFRAB+36                   ; 1217
                      065C      C3          058C      C3  9E 00238          MOVAB   NETBUF, NAFRAB+40                   ; 1218
                        03      A6            C0      8F  8A 0023F 15$:     BICB2   #192, IDENT+3                       ; 1225
02    A6                  0E                00          01  F0 00244      INSV    #1, #0, #14, IDENT+2                ; 1226
                                66                      04  B0 0024A      MOVW    #4, IDENT                           ; 1227
                                                        7E  7C 0024D      CLRQ    -(SP)                               ; 1228
                                                        7E  7C 0024F      CLRQ    -(SP)
                                                        7E  D4 00251      CLRL    -(SP)
                                                        66  DD 00253      PUSHL   IDENT
                      00000000G 00                      06  FB 00255      CALLS   #6, SYS$IDTOASC
                      00018292  8F                      50  D1 0025C      CMPL    STATUS, #98962                      ; 1229
                                                        04  12 00263      BNEQ    16$
                                         FC         A3  D4 00265          CLRL    RDB_EXISTS                          ; 1230
                                                        04  00268          RET
                          FC    A3                  01  D0 00269 16$:     MOVL    #1, RDB_EXISTS                      ; 1231
                                                        04  0026D          RET                                         ; 1233

; Routine Size:  622 bytes,    Routine Base:  $CODE$ + 0109
```

```
;  1149          1234  1 %sbttl 'add_uaf - insert new user record'
;  1150          1235  1 global routine add_uaf : novalue =
;  1151          1236  2 begin
;  1152          1237  2
;  1153          1238  2 !++
;  1154          1239  2 !
;  1155          1240  2 ! FUNCTIONAL DESCRIPTION:
;  1156          1241  2 !
;  1157          1242  2 !       Routine to add new user to authorization file.
;  1158          1243  2 !
;  1159          1244  2 ! INPUTS:
;  1160          1245  2 !
;  1161          1246  2 !       none
;  1162          1247  2 !
;  1163          1248  2 ! IMPLICIT INPUTS:
;  1164          1249  2 !
;  1165          1250  2 !       none
;  1166          1251  2 !
;  1167          1252  2 ! OUTPUTS:
;  1168          1253  2 !
;  1169          1254  2 !       none
;  1170          1255  2 !
;  1171          1256  2 ! IMPLICIT OUTPUTS:
;  1172          1257  2 !
;  1173          1258  2 !       none
;  1174          1259  2 !
;  1175          1260  2 ! ROUTINE VALUE:
;  1176          1261  2 !
;  1177          1262  2 !       none
;  1178          1263  2 !
;  1179          1264  2 ! SIDE EFFECTS:
;  1180          1265  2 !
;  1181          1266  2 !       A record is added to SYSUAF.DAT
;  1182          1267  2 !--
;  1183          1268  2
;  1184          1269  2 map
;  1185          1270  2         tokenptr        : ref vector [,byte];
;  1186          1271  2
;  1187          1272  2 local
;  1188          1273  2         user_dsc        : vector [2];   ! descriptor for username in record
;  1189          1274  2
;  1190          1275  2 !
;  1191          1276  2 ! Make sure a username was specified.
;  1192          1277  2 !
;  1193          1278  2
;  1194          1279  2 if not cli$present (sd_token1)
;  1195          1280  2 or not cli$get_value (sd_token1,tokendsc)
;  1196          1281  2 or .tokenlen eql 0
;  1197          1282  2 then return LIB$SIGNAL(UAF$_NOUSERNAME);
;  1198          1283  2
;  1199          1284  2 !
;  1200          1285  2 ! ADD must check that the username supplied is not too long.
;  1201          1286  2 !
;  1202          1287  2 !***if .tokenlen gtr  uaf$s_username
;  1203          1288  2 if .tokenlen gtr 12
;  1204          1289  2 then return LIB$SIGNAL(UAF$_NAMETOOBIG);
;  1205          1290  2
```

```
;  1206          1291  2 !
;  1207          1292  2 ! Make sure a legal username was entered, otherwise the account may not be
;  1208          1293  2 ! accessible via LOGIN or the Input Symbiont.
;  1209          1294  2 !
;  1210          1295  2 incru i to .tokenlen - 1
;  1211          1296  2 do
;  1212          1297  2     if ch$fail (ch$find_ch (.symbol_str<0,8>,
;  1213          1298  2                             symbol_str + 1,
;  1214          1299  2                             .tokenptr [.i]))
;  1215          1300  2     then return LIB$SIGNAL(UAF$_INVUSERNAME);
;  1216          1301  2 user_dsc[0] = .tokenlen;
;  1217          1302  2 user_dsc[1] = recbuf[uaf$t_username];
;  1218          1303  2
;  1219          1304  2 !
;  1220          1305  2 ! Move the default record to the current record buffer, so that
;  1221          1306  2 ! fields which are not entered will receive the default
;  1222          1307  2 ! value.  Then insert the username just entered.
;  1223          1308  2 !
;  1224          1309  2
;  1225          1310  2 ch$move (.default_size, default_record, recbuf);
;  1226          1311  2 ch$copy (.tokenlen, .tokenptr, ' ', uaf$s_username, recbuf[uaf$t_username]);
;  1227          1312  2
;  1228          1313  2 !
;  1229          1314  2 ! Call routine to fill in all supplied values.  Exit if any errors
;  1230          1315  2 ! were found.
;  1231          1316  2 !
;  1232          1317  2
;  1233          1318  2 pwd_flag = true;                          ! plan to supply a password
;  1234          1319  2 uafrab[rab$w_rsz] = .default_size;
;  1235          1320  2 uaf$gl_ctlmsk[uaf$v_add] = true;
;  1236          1321  2 if not update_record ()
;  1237          1322  2 then
;  1238          1323  3     begin
;  1239          1324  3     uaf$gl_ctlmsk[uaf$v_add] = false;
;  1240          1325  3     return;
;  1241          1326  2     end;
;  1242          1327  2
;  1243          1328  2 uaf$gl_ctlmsk[uaf$v_add] = false;
;  1244          1329  2
;  1245          1330  2 if .pwd_flag                             ! if no explicit password
;  1246          1331  2 then
;  1247          1332  3     begin
;  1248          1333  3     pwddsc[dsc$w_length] = .defpass<0,8>;
;  1249          1334  3     pwddsc[dsc$a_pointer] = defpass+1;
;  1250          1335  3     $gettim (timadr = time_buf);
;  1251          1336  3     recbuf[uaf$w_salt] = .time_buf<3*8,16>;
;  1252          1337  3     recbuf[uaf$b_encrypt] = encrypt;
;  1253          1338  3     lgi$hpwd (rec_encrypt_dsc, pwddsc, .recbuf[uaf$b_encrypt],
;  1254          1339  3                 .recbuf[uaf$w_salt], user_dsc);
;  1255          1340  2     end;
;  1256          1341  2
;  1257          1342  2 !
;  1258          1343  2 ! Now output the new record.
;  1259          1344  2 !
;  1260          1345  2
;  1261          1346  3 if rmsbad ($put (rab = uafrab))
;  1262          1347  2 then
```

```
;  1263          1348  2     if .rmserr eql rms$_dup
;  1264          1349  2     then return LIB$SIGNAL(UAF$_UAEERR)
;  1265          1350  2     else LIB$SIGNAL(UAF$_ADDERR, 0, .rmserr)
;  1266          1351  2 else
;  1267          1352  3     begin
;  1268          1353  3
;  1269          1354  3 !
;  1270          1355  3 ! Tell user that addition was successful.  Note that file was changed.
;  1271          1356  3 !
;  1272          1357  3     LIB$SIGNAL(UAF$_ADDMSG);
;  1273          1358  4     if (.uaf$gl_ctlmsk[uaf$v_cli]
;  1274          1359  4     and (not .uaf$gl_ctlmsk[uaf$v_clitables]))
;  1275          1360  3     then LIB$SIGNAL(UAF$_CLIWARNMSG);
;  1276          1361  3     security_audit (nsa$k_recid_sysuaf_add);
;  1277          1362  3     modify_flag = true;
;  1278          1363  4     if (cli$present ( sd_add_identifier ) and
;  1279          1364  4         .rdb_exists )
;  1280          1365  3     then
;  1281          1366  4         begin
;  1282          1367  4         !
;  1283          1368  4         ! Add the appropriate identifiers.
;  1284          1369  4         ! Set the resource attribute if /ATTRIB=RESOURCE was specified
;  1285          1370  4         ! and then add the appropriate identifiers to the rights data base
;  1286          1371  4         !
;  1287          1372  4         if cli$present (sd_attribresource)
;  1288          1373  4             then attributes = kgb$m_resource
;  1289          1374  4             else attributes = 0 ;
;  1290          1375  4         uaf$add_ident_recbuf () ;
;  1291          1376  3         end ;
;  1292          1377  2     end;
;  1293          1378  1 end;


                                                        .EXTRN  SYS$GETTIM

                                            0FFC 00000  .ENTRY  ADD_UAF, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-  ; 1235
                                                                R11
                                    5B 00000000G  00  9E 00002  MOVAB   LIB$SIGNAL, R11                        ;
                                    5A 00000000'  00  9E 00009  MOVAB   PWDDSC, R10                            ;
                                    59 00000000'  00  9E 00010  MOVAB   SD_TOKEN1, R9                          ;
                                    58 00000000'  00  9E 00017  MOVAB   TOKENDSC, R8                           ;
                                    5E            08  C2 0001E  SUBL2   #8, SP                                ;
                                                  59  DD 00021  PUSHL   R9                                   ; 1279
                          00000000G 00            01  FB 00023  CALLS   #1, CLI$PRESENT                      ;
                                    12            50  E9 0002A  BLBC    R0, 1$                               ;
                                                  58  DD 0002D  PUSHL   R8                                   ; 1280
                                                  59  DD 0002F  PUSHL   R9                                   ;
                          00000000G 00            02  FB 00031  CALLS   #2, CLI$GET_VALUE                    ;
                                    04            50  E9 00038  BLBC    R0, 1$                               ;
                                                  68  B5 0003B  TSTW    TOKENLEN                             ; 1281
                                                  08  12 0003D  BNEQ    2$                                   ;
                                        00000000G 8F  DD 0003F  1$:     PUSHL   #UAF$_NOUSERNAME             ; 1282
                                                  34  11 00045  BRB     6$                                   ;
                                    0C            68  B1 00047  2$:     CMPW    TOKENLEN, #12                ; 1288
                                                  08  1B 0004A  BLEQU   3$                                   ;
                                        00000000G 8F  DD 0004C  PUSHL   #UAF$_NAMETOOBIG                     ; 1289
```

```
                                                    27  11 00052                BRB     6$                                      ;
                                              53    68  3C 00054 3$:            MOVZWL  TOKENLEN, R3                            ; 1295
                                                    53  D7 00057                DECL    R3                                      ;
                                                    52  D4 00059                CLRL    I                                       ;
                                                    23  11 0005B                BRB     8$                                      ;
                                              51 01EC C9  9A 0005D 4$:          MOVZBL  SYMBOL_STR, R1                          ; 1297
                                              50 04  A8   D0 00062              MOVL    TOKENPTR, R0                            ; 1299
                  01ED  C9                    51        6240 3A 00066           LOCC    (I)[R0], R1, SYMBOL_STR+1               ;
                                                    02  12 0006D                BNEQ    5$                                      ;
                                                    51  D4 0006F                CLRL    R1                                      ;
                                                    51  D5 00071 5$:            TSTL    R1                                      ;
                                                    09  12 00073                BNEQ    7$                                      ;
                                        00000000G   8F  DD 00075                PUSHL   #UAF$_INVUSERNAME                       ; 1300
                                              00AF  31  0007B 6$:               BRW     11$                                     ;
                                                    52  D6 0007E 7$:            INCL    I                                       ; 1299
                                              53    52  D1 00080 8$:            CMPL    I, R3                                   ;
                                                    D8  1B 00083                BLEQU   4$                                      ;
                                              57    68  3C 00085                MOVZWL  TOKENLEN, R7                            ; 1301
                                              6E    57  D0 00088                MOVL    R7, USER_DSC                            ;
                                        04    AE  24  A8  9E 0008B              MOVAB   RECBUF+4, USER_DSC+4                    ; 1302
                                              56  FA78  CA  3C 00090            MOVZWL  DEFAULT_SIZE, R6                        ; 1310
                  20  A8        FA7C        CA        56  28 00095            MOVC3   R6, DEFAULT_RECORD, RECBUF              ;
                                              50  04  A8  D0 0009C              MOVL    TOKENPTR, R0                            ; 1311
      20                20                  60        57  2C 000A0            MOVC5   R7, (R0), #32, #32, RECBUF+4            ;
                                                  24  A8      000A5             ;                                               ;
                              0698  C8              01  D0 000A7              MOVL    #1, PWD_FLAG                            ; 1318
                              062A  C8              56  B0 000AC              MOVW    R6, UAFRAB+34                           ; 1319
                                A4  A8              02  88 000B1              BISB2   #2, UAF$GL_CTLMSK                       ; 1320
                        00000000G 00              00  FB 000B5              CALLS   #0, UPDATE_RECORD                       ; 1321
                                              05    50  E8 000BC              BLBS    R0, 9$                                  ;
                                A4  A8              02  8A 000BF              BICB2   #2, UAF$GL_CTLMSK                       ; 1324
                                                    04  000C3                RET                                             ; 1323
                                A4  A8              02  8A 000C4 9$:          BICB2   #2, UAF$GL_CTLMSK                       ; 1328
                                              3B    0698 C8  E9 000C8          BLBC    PWD_FLAG, 10$                           ; 1330
                                              6A    0120 C9  9B 000CD          MOVZBW  DEFPASS, PWDDSC                         ; 1333
                                        04    AA    0121 C9  9E 000D2          MOVAB   DEFPASS+1, PWDDSC+4                     ; 1334
                                              0690  C8  9F 000D8              PUSHAB  TIME_BUF                                ; 1335
                              00000000G       00        01  FB 000DC          CALLS   #1, SYS$GETTIM                          ;
                                0186  C8      0693  C8  B0 000E3              MOVW    TIME_BUF+3, RECBUF+358                  ; 1336
                                0188  C8            02  90 000EA              MOVB    #2, RECBUF+360                          ; 1337
                                                    5E  DD 000EF              PUSHL   SP                                      ; 1338
                                              7E    0186 C8  3C 000F1          MOVZWL  RECBUF+358, -(SP)                       ; 1339
                                              7E    0188 C8  9A 000F6          MOVZBL  RECBUF+360, -(SP)                       ; 1338
                                                    5A  DD 000FB              PUSHL   R10                                     ;
                                              01E4  C9  9F 000FD              PUSHAB  REC_ENCRYPT_DSC                         ;
                              00000000G       00        05  FB 00101          CALLS   #5, LGI$HPWD                            ;
                                              0608  C8  9F 00108 10$:          PUSHAB  UAFRAB                                  ; 1346
                              00000000G       00        01  FB 0010C          CALLS   #1, SYS$PUT                             ;
                                      18      A8        50  D0 00113          MOVL    R0, RMSERR                              ;
                                              25        50  E8 00117          BLBS    R0, 13$                                 ;
                                              50    18  A8  D0 0011A          MOVL    RMSERR, R0                              ; 1348
                              000184EC        8F        50  D1 0011E          CMPL    R0, #99564                              ;
                                                        0A  12 00125          BNEQ    12$                                     ;
                                        00000000G       8F  DD 00127          PUSHL   #UAF$_UAEERR                            ; 1349
                                              6B            01  FB 0012D 11$:   CALLS   #1, LIB$SIGNAL                         ;
                                                            04  00130          RET                                             ;
                                                            50  DD 00131 12$:   PUSHL   R0                                      ; 1350
```

```
                                                7E  D4 00133          CLRL    -(SP)                                                   ;
                                 00000000G      8F  DD 00135          PUSHL   #UAF$_ADDERR                                            ;
                                         6B     03  FB 0013B          CALLS   #3, LIB$SIGNAL                                          ;
                                                04  0013E          RET                                                             ; 1348
                                 00000000G      8F  DD 0013F 13$:     PUSHL   #UAF$_ADDMSG                                            ; 1357
                                         6B     01  FB 00145          CALLS   #1, LIB$SIGNAL                                          ;
                              0E         A4     A8  03  E1 00148          BBC     #3, UAF$GL_CTLMSK, 14$                          ; 1358
                              09         A4     A8  04  E0 0014D          BBS     #4, UAF$GL_CTLMSK, 14$                          ; 1359
                                 00000000G      8F  DD 00152          PUSHL   #UAF$_CLIWARNMSG                                        ; 1360
                                         6B     01  FB 00158          CALLS   #1, LIB$SIGNAL                                          ;
                                 00010002       8F  DD 0015B 14$:     PUSHL   #65538                                                  ; 1361
                                00000000V       00  01  FB 00161          CALLS   #1, SECURITY_AUDIT                                      ;
                                     F624       CA  01  D0 00168          MOVL    #1, MODIFY_FLAG                                         ; 1362
                                               44   A9  9F 0016D          PUSHAB  SD_ADD_IDENTIFIER                                       ; 1363
                                00000000G      00  01  FB 00170          CALLS   #1, CLI$PRESENT                                         ;
                                               28  50  E9 00177          BLBC    R0, 17$                                                 ;
                                               24  14  A8  E9 0017A          BLBC    RDB_EXISTS, 17$                                 ; 1364
                                             F5D8  CA  9F 0017E          PUSHAB  SD_ATTRIBRESOURCE                                       ; 1372
                                00000000G      00  01  FB 00182          CALLS   #1, CLI$PRESENT                                         ;
                                               09  50  E9 00189          BLBC    R0, 15$                                                 ;
                                00000000G      00  01  D0 0018C          MOVL    #1, ATTRIBUTES                                          ; 1373
                                               06  11  00193          BRB     16$                                                     ;
                                 00000000G      00  D4 00195 15$:     CLRL    ATTRIBUTES                                              ; 1374
                                00000000G      00  00  FB 0019B 16$:     CALLS   #0, UAF$ADD_IDENT_RECBUF                                ; 1375
                                                   04  001A2 17$:     RET                                                             ; 1378

; Routine Size:  419 bytes,    Routine Base:  $CODE$ + 0377
```

```
;  1295     1379  1 %sbttl 'add_proxy - insert new proxy record'
;  1296     1380  1 global routine add_proxy : novalue =
;  1297     1381  2 begin
;  1298     1382  2
;  1299     1383  2 !++
;  1300     1384  2 !
;  1301     1385  2 ! FUNCTIONAL DESCRIPTION:
;  1302     1386  2 !
;  1303     1387  2 !       This routine adds an entry to the NETUAF.DAT Proxy Login File
;  1304     1388  2 !
;  1305     1389  2 ! INPUTS:
;  1306     1390  2 !
;  1307     1391  2 !       none
;  1308     1392  2 !
;  1309     1393  2 ! OUTPUTS:
;  1310     1394  2 !
;  1311     1395  2 !       none
;  1312     1396  2 !
;  1313     1397  2 ! IMPLICIT INPUTS:
;  1314     1398  2 !
;  1315     1399  2 !       TOKENLEN, TOKENPTR
;  1316     1400  2 !
;  1317     1401  2 ! IMPLICIT OUTPUTS:
;  1318     1402  2 !
;  1319     1403  2 !       none
;  1320     1404  2 !
;  1321     1405  2 ! ROUTINE VALUE:
;  1322     1406  2 !
;  1323     1407  2 !       none
;  1324     1408  2 !
;  1325     1409  2 ! SIDE EFFECTS:
;  1326     1410  2 !
;  1327     1411  2 !       A record is added to NETUAF.DAT
;  1328     1412  2 !
;  1329     1413  2 !--
;  1330     1414  2
;  1331     1415  2 local
;  1332     1416  2         node_len,
;  1333     1417  2         node_ptr,
;  1334     1418  2         remuser_len,
;  1335     1419  2         remuser_ptr,
;  1336     1420  2         locuser_len,
;  1337     1421  2         locuser_ptr;
;  1338     1422  2
;  1339     1423  2 !
;  1340     1424  2 ! Can't do anything if there is no NETUAF.DAT...
;  1341     1425  2 !
;  1342     1426  2 if not .netuaf_exists
;  1343     1427  2 then return LIB$SIGNAL(UAF$_NAFDNE);
;  1344     1428  2
;  1345     1429  2 !
;  1346     1430  2 ! Clear NETUAF.DAT buffer
;  1347     1431  2 !
;  1348     1432  2 ch$fill (' ', naf$c_length, netbuf);
;  1349     1433  2
;  1350     1434  2 !
;  1351     1435  2 ! Retrieve token from command line
```

```
;  1352     1436  2 !
;  1353     1437  2 cli$get_value (sd_token1, tokendsc);
;  1354     1438  2
;  1355     1439  2 !
;  1356     1440  2 ! Make sure entry is in proper node::remoteuser format
;  1357     1441  2 !
;  1358     1442  2 if not remote_parse (node_ptr, node_len, remuser_ptr, remuser_len)
;  1359     1443  2 then return;
;  1360     1444  2 !
;  1361     1445  2 ! Fill in NETBUF with new remotename field
;  1362     1446  2 !
;  1363     1447  2 ch$copy (.node_len, .node_ptr, ' ', naf$s_node, netbuf[naf$t_node]);
;  1364     1448  2 ch$copy (.remuser_len, .remuser_ptr, ' ', naf$s_remuser, netbuf[naf$t_remuser]);
;  1365     1449  2
;  1366     1450  2 !
;  1367     1451  2 ! Now get second token, the local user name
;  1368     1452  2 !
;  1369     1453  2 cli$get_value (sd_token2, tokendsc);
;  1370     1454  2
;  1371     1455  2 locuser_len = .tokenlen;
;  1372     1456  2 locuser_ptr = .tokenptr;
;  1373     1457  2 !***if .locuser_len gtru naf$s_localuser
;  1374     1458  2 if .locuser_len gtru 12
;  1375     1459  2 then return LIB$SIGNAL(UAF$_NAMETOOBIG);
;  1376     1460  2
;  1377     1461  2 !
;  1378     1462  2 ! If local name is *, then use same name as remote user
;  1379     1463  2 !
;  1380     1464  2 if .tokenlen eql 1  and  . (.tokenptr)<0,8> eql '*'
;  1381     1465  2 then
;  1382     1466  3     begin
;  1383     1467  3     locuser_len = .remuser_len;
;  1384     1468  3     ch$move (naf$s_remuser, netbuf[naf$t_remuser], netbuf[naf$t_localuser]);
;  1385     1469  3     end
;  1386     1470  2 !
;  1387     1471  2 ! Otherwise just copy into localuser field in NETBUF
;  1388     1472  2 !
;  1389     1473  2 else
;  1390     1474  2     ch$copy (.locuser_len, .locuser_ptr, ' ',
;  1391     1475  2              naf$s_localuser, netbuf[naf$t_localuser]);
;  1392     1476  2
;  1393     1477  2 !
;  1394     1478  2 ! Make sure that the local user does indeed exist in SYSUAF.DAT
;  1395     1479  2 ! (unless local user is *)
;  1396     1480  2 !
;  1397     1481  2 if not locate_user (.locuser_len, netbuf[naf$t_localuser], 0)
;  1398     1482  3 and not (.locuser_len eql 1 and . (netbuf[naf$t_localuser])<0,8> eql '*')
;  1399     1483  2 then return LIB$SIGNAL(UAF$_BADUSR, 2, .locuser_len, netbuf[naf$t_localuser]);
;  1400     1484  2
;  1401     1485  2 nafrab[rab$w_rsz] = naf$c_length;
;  1402     1486  2
;  1403     1487  2 !
;  1404     1488  2 ! Add NETUAF.DAT record
;  1405     1489  2 !
;  1406     1490  3 if rmsbad ($put (rab = nafrab))
;  1407     1491  2 then
;  1408     1492  3     begin
```

```
;  1409          1493  3     if .rmserr eql rms$_dup
;  1410          1494  3     then
;  1411          1495  3         return LIB$SIGNAL(UAF$_NAFUAEERR)
;  1412          1496  3     else
;  1413          1497  3         LIB$SIGNAL(UAF$_NAFADDERR, 0, .rmserr)
;  1414          1498  3     end
;  1415          1499  2 else
;  1416          1500  2     begin
;  1417          1501  2     LIB$SIGNAL(UAF$_NAFADDMSG);
;  1418          1502  2     security_audit (nsa$k_recid_netuaf_add);
;  1419          1503  2     end;
;  1420          1504  2
;  1421          1505  2 netuaf_modified = true;
;  1422          1506  1 end;


                                                07FC 00000          .ENTRY  ADD_PROXY, Save R2,R3,R4,R5,R6,R7,R8,R9,R10 ; 1380
                                    5A 00000000G   00   9E 00002    MOVAB   CLI$GET_VALUE, R10                          ;
                                    59 00000000G   00   9E 00009    MOVAB   LIB$SIGNAL, R9                              ;
                                    58 00000000'   00   9E 00010    MOVAB   NETBUF+64, R8                               ;
                                    5E             10   C2 00017    SUBL2   #16, SP                                     ;
                                    08      FA2C   C8   E8 0001A    BLBS    NETUAF_EXISTS, 1$                           ; 1426
                                       00000000G   8F   DD 0001F    PUSHL   #UAF$_NAFDNE                                ; 1427
                                                   5F   11 00025    BRB     3$                                          ;
    0064   8F             20                    6E      2C 00027 1$:    MOVC5   #0, (SP), #32, #100, NETBUF                 ; 1432
                                                C0   A8    0002E                                                        ;
                                          FA1C  C8   9F 00030         PUSHAB  TOKENDSC                                    ; 1437
                                       00000000'  00   9F 00034         PUSHAB  SD_TOKEN1                                   ;
                                    6A            02   FB 0003A         CALLS   #2, CLI$GET_VALUE                           ;
                                                  5E   DD 0003D         PUSHL   SP                                          ; 1442
                                                08    AE   9F 0003F     PUSHAB  REMUSER_PTR                                 ;
                                                10    AE   9F 00042     PUSHAB  NODE_LEN                                    ;
                                                18    AE   9F 00045     PUSHAB  NODE_PTR                                    ;
                                          00000000V   00   04   FB 00048    CALLS   #4, REMOTE_PARSE                         ;
                                                      01   50   E8 0004F    BLBS    R0, 2$                                   ;
                                                           04   00052    RET                                              ;
    20             20          0C    BE        08    AE   2C 00053 2$:    MOVC5   NODE_LEN, @NODE_PTR, #32, #32, NETBUF    ; 1447
                                                C0   A8    0005A                                                        ;
    20             20          04    BE              6E   2C 0005C         MOVC5   REMUSER_LEN, @REMUSER_PTR, #32, #32, -   ; 1448
                                                E0   A8    00062                    NETBUF+32                           ;
                                          FA1C  C8   9F 00064         PUSHAB  TOKENDSC                                    ; 1453
                                       00000000'  00   9F 00068         PUSHAB  SD_TOKEN2                                   ;
                                    6A            02   FB 0006E         CALLS   #2, CLI$GET_VALUE                           ;
                                    56      FA1C  C8   3C 00071         MOVZWL  TOKENLEN, LOCUSER_LEN                       ; 1455
                                    57      FA20  C8   D0 00076         MOVL    TOKENPTR, LOCUSER_PTR                       ; 1456
                                    0C            56   D1 0007B         CMPL    LOCUSER_LEN, #12                            ; 1458
                                                  08   1B 0007E         BLEQU   4$                                          ;
                                       00000000G  8F   DD 00080         PUSHL   #UAF$_NAMETOOBIG                            ; 1459
                                                  77   11 00086 3$:     BRB     9$                                          ;
                                    01      FA1C  C8   B1 00088 4$:     CMPW    TOKENLEN, #1                                ; 1464
                                                  14   12 0008D         BNEQ    5$                                          ;
                                    50      FA20  C8   D0 0008F         MOVL    TOKENPTR, R0                                ;
                                    2A            60   91 00094         CMPB    (R0), #42                                   ;
                                                  0A   12 00097         BNEQ    5$                                          ;
```

```
;  add_proxy - insert new proxy record

                                 56                      6E  D0 00099          MOVL    REMUSER_LEN, LOCUSER_LEN                        ; 1467
                             68      E0  A8              20  28 0009C          MOVC3   #32, NETBUF+32, NETBUF+64                       ; 1468
                                         67              06  11 000A1          BRB     6$                                              ; 1464
                         20      20                      56  2C 000A3 5$:      MOVC5   LOCUSER_LEN, (LOCUSER_PTR), #32, #32, -         ; 1475
                                                                 68  000A8              NETBUF+64
                                                         7E  D4 000A9 6$:       CLRL    -(SP)                                           ; 1481
                                             0140        8F  BB 000AB          PUSHR   #^M<R6,R8>
                                 00000000V   00          03  FB 000AF          CALLS   #3, LOCATE_USER
                                             1A          50  E8 000B6          BLBS    R0, 8$
                                             01          56  D1 000B9          CMPL    LOCUSER_LEN, #1                                 ; 1482
                                             2A          05  12 000BC          BNEQ    7$
                                                         68  91 000BE          CMPB    NETBUF+64, #42
                                                         10  13 000C1          BEQL    8$
                                             0140        8F  BB 000C3 7$:       PUSHR   #^M<R6,R8>                                      ; 1483
                                                         02  DD 000C7          PUSHL   #2
                                         00000000G       8F  DD 000C9          PUSHL   #UAF$_BADUSR
                                             69          04  FB 000CF          CALLS   #4, LIB$SIGNAL
                                                         04  000D2          RET
                                   008A      C8      64  8F  9B 000D3 8$:       MOVZBW  #100, NAFRAB+34                                 ; 1485
                                             68      A8  9F  000D9          PUSHAB  NAFRAB                                          ; 1490
                                 00000000G   00          01  FB 000DC          CALLS   #1, SYS$PUT
                                     FA34    C8          50  D0 000E3          MOVL    R0, RMSERR
                                             27          50  E8 000E8          BLBS    R0, 11$
                                             50      FA34    C8  D0 000EB      MOVL    RMSERR, R0                                      ; 1493
                                 000184EC    8F          50  D1 000F0          CMPL    R0, #99564
                                                         0A  12 000F7          BNEQ    10$
                                         00000000G       8F  DD 000F9          PUSHL   #UAF$_NAFUAEERR                                 ; 1495
                                             69          01  FB 000FF 9$:       CALLS   #1, LIB$SIGNAL
                                                         04  00102          RET
                                                         50  DD 00103 10$:      PUSHL   R0                                              ; 1497
                                                         7E  D4 00105          CLRL    -(SP)
                                         00000000G       8F  DD 00107          PUSHL   #UAF$_NAFADDERR
                                             69          03  FB 0010D          CALLS   #3, LIB$SIGNAL
                                                         16  11 00110          BRB     12$                                             ; 1492
                                         00000000G       8F  DD 00112 11$:      PUSHL   #UAF$_NAFADDMSG                                 ; 1501
                                             69          01  FB 00118          CALLS   #1, LIB$SIGNAL
                                         00010003        8F  DD 0011B          PUSHL   #65539                                          ; 1502
                                 00000000V   00          01  FB 00121          CALLS   #1, SECURITY_AUDIT
                                 00000000'   00          01  D0 00128 12$:      MOVL    #1, NETUAF_MODIFIED                             ; 1505
                                                         04  0012F          RET                                                     ; 1506

; Routine Size:  304 bytes,    Routine Base:  $CODE$ + 051A
```

```
;  1424          1507  1 %sbttl 'remote_parse - parses "node::remoteuser"'
;  1425          1508  1 routine remote_parse (node_ptr, node_len, remuser_ptr, remuser_len) =
;  1426          1509  2 begin
;  1427          1510  2
;  1428          1511  2 !++
;  1429          1512  2 !
;  1430          1513  2 ! FUNCTIONAL DESCRIPTION:
;  1431          1514  2 !
;  1432          1515  2 !       This routine parses a remote user specification in the form
;  1433          1516  2 !          node::remuser, and returns the two components by lengths
;  1434          1517  2 !          and pointers to the strings
;  1435          1518  2 !
;  1436          1519  2 ! INPUTS:
;  1437          1520  2 !
;  1438          1521  2 !       node_ptr - returned as pointer to nodename
;  1439          1522  2 !       node_len -      "       " length of nodename
;  1440          1523  2 !       remuser_ptr - returned as pointer to remote user name
;  1441          1524  2 !       remuser_len -      "       "    length of remote user name
;  1442          1525  2 !
;  1443          1526  2 ! IMPLICIT INPUTS:
;  1444          1527  2 !
;  1445          1528  2 !       TOKENLEN and TOKENPTR - the remote user specification is assumed
;  1446          1529  2 !          to have just been fetched from the command line
;  1447          1530  2 !
;  1448          1531  2 ! OUTPUTS:
;  1449          1532  2 !
;  1450          1533  2 !       none
;  1451          1534  2 !
;  1452          1535  2 ! ROUTINE VALUE:
;  1453          1536  2 !
;  1454          1537  2 !       TRUE if parsed successfully
;  1455          1538  2 !       FALSE if error encountered in parsing
;  1456          1539  2 !
;  1457          1540  2 !--
;  1458          1541  2
;  1459          1542  2 map
;  1460          1543  2         dbl_colon       : vector;
;  1461          1544  2
;  1462          1545  2 local
;  1463          1546  2         dbl_colon_ptr;
;  1464          1547  2
;  1465          1548  2 !
;  1466          1549  2 ! Better be able to find a double colon in the remotename...
;  1467          1550  2 !
;  1468          1551  2 dbl_colon_ptr = ch$find_sub (.tokenlen, .tokenptr, 2, .dbl_colon [1]);
;  1469          1552  2
;  1470          1553  2 if .dbl_colon_ptr eql 0                           ! no double colon found
;  1471          1554  2 or .dbl_colon_ptr eql .tokenptr                   ! no node found
;  1472          1555  3 or .dbl_colon_ptr eql (.tokenptr + .tokenlen - 2) ! no remote user found
;  1473          1556  2 then return LIB$SIGNAL(UAF$_BADNODFORM);
;  1474          1557  2
;  1475          1558  2 !
;  1476          1559  2 ! Determine node length and pointer
;  1477          1560  2 !
;  1478          1561  2 .node_len = .dbl_colon_ptr - .tokenptr;
;  1479          1562  2 .node_ptr = .tokenptr;
;  1480          1563  2
```

```
;  1481          1564  2 !
;  1482          1565  2 ! Make sure node name isn't too long
;  1483          1566  2 !
;  1484          1567  2 !***if . (.node_len) gtru naf$s_node
;  1485          1568  2 if . (.node_len) gtru 6
;  1486          1569  2 then return LIB$SIGNAL(UAF$_NODTOOBIG);
;  1487          1570  2
;  1488          1571  2 !
;  1489          1572  2 ! Determine remote username length and pointer
;  1490          1573  2 !
;  1491          1574  2 .remuser_len = .tokenlen - .(.node_len) - 2;
;  1492          1575  2 .remuser_ptr = .dbl_colon_ptr + 2;
;  1493          1576  2
;  1494          1577  2 !
;  1495          1578  2 ! And make sure name isn't too long
;  1496          1579  2 !
;  1497          1580  2 !***if . (.remuser_len) gtru naf$s_remuser
;  1498          1581  2 if . (.remuser_len) gtru 12
;  1499          1582  2 then return LIB$SIGNAL(UAF$_NAMETOOBIG);
;  1500          1583  2
;  1501          1584  2 return true;
;  1502          1585  1 end;


                                    007C 00000 REMOTE_PARSE:
                                                .WORD   Save R2,R3,R4,R5,R6                     ;  1508
                          56 00000000'  00  9E 00002    MOVAB   TOKENLEN, R6
                          55            66  3C 00009    MOVZWL  TOKENLEN, R5                    ;  1551
                          54      04    A6  D0 0000C    MOVL    TOKENPTR, R4
                          50 00000000'  00  D0 00010    MOVL    DBL_COLON+4, R0
64                55      60            02  39 00017    MATCHC  #2, (R0), R5, (R4)
                                        03  13 0001C    BEQL    1$
                          53            02  D0 0001E    MOVL    #2, R3
                          53            02  C2 00021 1$: SUBL2   #2, R3
                                        0F  13 00024    BEQL    2$                              ;  1553
                          54            53  D1 00026    CMPL    DBL_COLON_PTR, R4               ;  1554
                                        0A  13 00029    BEQL    2$
                          50      FE A544  9E 0002B    MOVAB   -2(R5)[R4], R0                  ;  1555
                          50            53  D1 00030    CMPL    DBL_COLON_PTR, R0
                                        08  12 00033    BNEQ    3$
                       00000000G        8F  DD 00035 2$: PUSHL   #UAF$_BADNODFORM                ;  1556
                                        38  11 00038    BRB     5$
                          50      04    A6  D0 0003D 3$: MOVL    TOKENPTR, R0                    ;  1561
            08    BC      53            50  C3 00041    SUBL3   R0, DBL_COLON_PTR, @NODE_LEN
                  04    BC              50  D0 00046    MOVL    R0, @NODE_PTR                   ;  1562
                  06      08    BC      D1  0004A    CMPL    @NODE_LEN, #6                   ;  1568
                                        08  1B 0004E    BLEQU   4$
                       00000000G        8F  DD 00050    PUSHL   #UAF$_NODTOOBIG                 ;  1569
                                        1D  11 00056    BRB     5$
                          50            66  3C 00058 4$: MOVZWL  TOKENLEN, R0                    ;  1574
                          50      08    BC  C2 0005B    SUBL2   @NODE_LEN, R0
                  10    BC        FE    A0  9E 0005F    MOVAB   -2(R0), @REMUSER_LEN
                  0C    BC        02    A3  9E 00064    MOVAB   2(R3), @REMUSER_PTR             ;  1575
                  0C      10    BC      D1  00069    CMPL    @REMUSER_LEN, #T2               ;  1581
```

```
                                 0E  1B 0006D          BLEQU   6$                                      ;
                    00000000G    8F  DD 0006F          PUSHL   #UAF$_NAMETOOBIG                        ; 1582
          00000000G  00          01  FB 00075 5$:      CALLS   #1, LIB$SIGNAL                          ;
                                 04  0007C            RET                                             ;
                     50          01  D0 0007D 6$:      MOVL    #1, R0                                  ; 1584
                                 04  00080            RET                                             ; 1585

; Routine Size:  129 bytes,    Routine Base:  $CODE$ + 064A
```

```
;  1504          1586  1 %sbttl 'copy_uaf - copy user record'
;  1505          1587  1 global routine copy_uaf  =
;  1506          1588  2 begin
;  1507          1589  2
;  1508          1590  2 !++
;  1509          1591  2 !
;  1510          1592  2 ! FUNCTIONAL DESCRIPTION:
;  1511          1593  2 !
;  1512          1594  2 !       Routine to copy a user authorization record, giving the
;  1513          1595  2 !       new authorization record a different name.
;  1514          1596  2 !
;  1515          1597  2 ! INPUTS:
;  1516          1598  2 !
;  1517          1599  2 !       none
;  1518          1600  2 !
;  1519          1601  2 ! IMPLICIT INPUTS:
;  1520          1602  2 !
;  1521          1603  2 !       none
;  1522          1604  2 !
;  1523          1605  2 ! OUTPUTS:
;  1524          1606  2 !
;  1525          1607  2 !       none
;  1526          1608  2 !
;  1527          1609  2 ! ROUTINE VALUE:
;  1528          1610  2 !
;  1529          1611  2 !        false if the copy fails;
;  1530          1612  2 !        true  if the copy succeeds.
;  1531          1613  2 !
;  1532          1614  2 ! SIDE EFFECTS:
;  1533          1615  2 !
;  1534          1616  2 !       A user record is added.
;  1535          1617  2 !
;  1536          1618  2 !
;  1537          1619  2 !--
;  1538          1620  2
;  1539          1621  2 local
;  1540          1622  2         status,
;  1541          1623  2         flag,
;  1542          1624  2         lock_rec,
;  1543          1625  2         def_sys,
;  1544          1626  2         old_user_buffer : vector [uaf$s_username, byte];
;  1545          1627  2
;  1546          1628  2 map
;  1547          1629  2         tokenptr        : ref vector [,byte];
;  1548          1630  2
;  1549          1631  2 uaf$gl_ctlmsk[uaf$v_copy] = not .uaf$gl_ctlmsk[uaf$v_rename];
;  1550          1632  2 !
;  1551          1633  2 ! If this is a COPY directly from the UAF> prompt, the authorization
;  1552          1634  2 ! record need not be locked, and the default and system records may be
;  1553          1635  2 ! copied.  HOWEVER, if this COPY is part of a RENAME operation, the record
;  1554          1636  2 ! must be locked, and the default and system records may not be renamed.
;  1555          1637  2 ! (The RENAME operation is similar to the COPY operation except that
;  1556          1638  2 ! the original record is REMOVE'd.  COPY leaves both records.)
;  1557          1639  2 !
;  1558          1640  2 if not .uaf$gl_ctlmsk[uaf$v_rename]
;  1559          1641  2 then
;  1560          1642  3     begin
```

```
; 1561        1643  3      lock_rec = false;                    ! A COPY operation
; 1562        1644  3      def_sys  = true;
; 1563        1645  3      flag = false;
; 1564        1646  3      end
; 1565        1647  2  else
; 1566        1648  3      begin
; 1567        1649  3      lock_rec = true;                     ! A RENAME operation
; 1568        1650  3      def_sys  = false;
; 1569        1651  3      flag = true;
; 1570        1652  2      end;
; 1571        1653  2
; 1572        1654  2  !
; 1573        1655  2  !  Place record to be copied into RECBUF
; 1574        1656  2  !  (If the third argument is true, the call to GET_USER_RECORD
; 1575        1657  2  !  is part of a RENAME operation, and the first token should be saved.
; 1576        1658  2  !  If the third argument is false, the call is part of a COPY
; 1577        1659  2  !  operation, and the first token need not be saved.)
; 1578        1660  2  !
; 1579        1661  2
; 1580        1662  2  if get_user_record (.lock_rec, .def_sys, .flag)
; 1581        1663  2  then                                     ! Make sure a username was
; 1582        1664  3      begin                                ! specified
; 1583        1665  3      if not cli$present (sd_token2)
; 1584        1666  3      or not cli$get_value (sd_token2, tokendsc)
; 1585        1667  3      or .tokenlen eql 0
; 1586        1668  3      then return LIB$SIGNAL(UAF$_NOUSERNAME);
; 1587        1669  3
; 1588        1670  3  !
; 1589        1671  3  !  Make sure that the new username isn't too long
; 1590        1672  3  !
; 1591        1673  3  !***     if .tokenlen  gtru uaf$s_username
; 1592        1674  3      if .tokenlen  gtru 12
; 1593        1675  3      then LIB$SIGNAL(UAF$_NAMETOOBIG);
; 1594        1676  3
; 1595        1677  3  !
; 1596        1678  3  !  Make sure that the new username is legal
; 1597        1679  3  !
; 1598        1680  3      incru i to .tokenlen - 1
; 1599        1681  3      do
; 1600        1682  3          if ch$fail (ch$find_ch (.symbol_str<0,8>,
; 1601        1683  3                                  symbol_str + 1,
; 1602        1684  3                                 .tokenptr [.i]))
; 1603        1685  3          then return LIB$SIGNAL(UAF$_INVUSERNAME);
; 1604        1686  3
; 1605        1687  3  !
; 1606        1688  3  !  If this is a rename save the new user name
; 1607        1689  3  !
; 1608        1690  3  if .uaf$gl_ctlmsk[uaf$v_rename]
; 1609        1691  3  then
; 1610        1692  4      begin
; 1611        1693  4      ch$move (.tokenlen, .tokenptr, newusername);
; 1612        1694  4      newuserlen = .tokenlen;
; 1613        1695  3      end ;
; 1614        1696  3
; 1615        1697  3  !
; 1616        1698  3  !  Place the new username in RECBUF, but save the old username first
; 1617        1699  3  !
```

```
; 1618          1700  3     ch$move (uaf$s_username, recbuf[uaf$t_username], old_user_buffer);
; 1619          1701  3     ch$copy (.tokenlen, .tokenptr, ' ',
; 1620          1702  3             uaf$s_username, recbuf[uaf$t_username]);
; 1621          1703  3     pwd_flag = true;
; 1622          1704  3
; 1623          1705  3     status = update_record ();
; 1624          1706  3     if not .status
; 1625          1707  3     then return false;
; 1626          1708  3
; 1627          1709  3 !
; 1628          1710  3 ! If this is a copy operation then zero fill the last login info
; 1629          1711  3 !
; 1630          1712  3 if not .uaf$gl_ctlmsk[uaf$v_rename]
; 1631          1713  3 then
; 1632          1714  4     begin
; 1633          1715  4     recbuf[uaf$w_logfails] = 0 ;
; 1634          1716  4     ch$fill ( 0, uaf$s_lastlogin_i, recbuf[uaf$q_lastlogin_i] ) ;
; 1635          1717  4     ch$fill ( 0, uaf$s_lastlogin_n, recbuf[uaf$q_lastlogin_n] ) ;
; 1636          1718  3     end ;
; 1637          1719  3
; 1638          1720  3 !
; 1639          1721  3 ! Now output the new record
; 1640          1722  3 !
; 1641          1723  4     if rmsbad ($put (rab = uafrab))
; 1642          1724  3     then
; 1643          1725  4         begin
; 1644          1726  4         if .rmserr eql rms$_dup
; 1645          1727  4         then
; 1646          1728  4             return LIB$SIGNAL(UAF$_UAEERR)      ! username already exists
; 1647          1729  4         else
; 1648          1730  5             begin
; 1649          1731  5             LIB$SIGNAL(UAF$_ADDERR, 0, .rmserr);
; 1650          1732  5             return false;
; 1651          1733  5             end
; 1652          1734  4         end
; 1653          1735  3     else
; 1654          1736  3 !
; 1655          1737  3 ! The copy was successful -- tell the user and set modify flag
; 1656          1738  3 !
; 1657          1739  4         begin
; 1658          1740  4         modify_flag = true;
; 1659          1741  5         security_audit ((if not .uaf$gl_ctlmsk[uaf$v_rename]
; 1660          1742  5                          then nsa$k_recid_sysuaf_cop
; 1661          1743  4                          else nsa$k_recid_sysuaf_ren),
; 1662          1744  4                         old_user_buffer);
; 1663          1745  4         if not .uaf$gl_ctlmsk[uaf$v_rename]
; 1664          1746  4         then
; 1665          1747  5             begin
; 1666          1748  5             LIB$SIGNAL(UAF$_COPMSG);
; 1667          1749  6             if (.uaf$gl_ctlmsk[uaf$v_cli]
; 1668          1750  6             and not .uaf$gl_ctlmsk[uaf$v_clitables])
; 1669          1751  5             and .uaf$gl_ctlmsk[uaf$v_clitab_pres]
; 1670          1752  5             then LIB$SIGNAL(UAF$_CLIWARNMSG);
; 1671          1753  5             !
; 1672          1754  5             ! Since passwords are folded in with the username, passwords for
; 1673          1755  5             ! COPYed records will no longer work--warn the user
; 1674          1756  5             !
```

```
;  1675          1757  5              if .pwd_flag
;  1676          1758  5              then LIB$SIGNAL(UAF$_DEFPWD);
;  1677          1759  5              uaf$gl_ctlmsk[uaf$v_copy] = false;
;  1678          1760  6              if (cli$present ( sd_add_identifier ) and
;  1679          1761  6                  .rdb_exists )
;  1680          1762  5              then
;  1681          1763  6                  begin
;  1682          1764  6                  !
;  1683          1765  6                  ! Set the resource attribute if /ATTRIB=RESOURCE was specified
;  1684          1766  6                  ! and then add the appropriate identifiers to the rights data base
;  1685          1767  6                  !
;  1686          1768  6                  if cli$present (sd_attribresource)
;  1687          1769  6                      then attributes = kgb$m_resource
;  1688          1770  6                      else attributes = 0 ;
;  1689          1771  6                  uaf$add_ident_recbuf () ;
;  1690          1772  5                  end;
;  1691          1773  4              end;
;  1692          1774  3          end ;
;  1693          1775  3      end
;  1694          1776  3
;  1695          1777  3  !
;  1696          1778  3  ! The attempt to GET_USER_RECORD failed...
;  1697          1779  3  !
;  1698          1780  2  else return false;
;  1699          1781  2
;  1700          1782  2  !
;  1701          1783  2  ! If we get here, everything succeeded -- return true
;  1702          1784  2  !
;  1703          1785  2  return true;
;  1704          1786  2
;  1705          1787  1  end;


                                                OFFC 00000          .ENTRY  COPY_UAF, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-  ; 1587
                                                                            R11                                           ;
                                    5B 00000000G   00   9E 00002    MOVAB   CLI$PRESENT, R11                              ;
                                    5A 00000000'   00   9E 00009    MOVAB   NEWUSERNAME, R10                              ;
                                    59 00000000'   00   9E 00010    MOVAB   SD_TOKEN2, R9                                 ;
                                    58 00000000G   00   9E 00017    MOVAB   LIB$SIGNAL, R8                                ;
                                    57 00000000'   00   9E 0001E    MOVAB   UAF$GL_CTLMSK, R7                             ;
                                    5E             20   C2 00025    SUBL2   #32, SP                                       ;
        50              67          01             00   EF 00028    EXTZV   #0, #1, UAF$GL_CTLMSK, R0                     ; 1631
                                    50             50   92 0002D    MCOMB   R0, R0                                        ;
        67              01          02             50   F0 00030    INSV    R0, #2, #1, UAF$GL_CTLMSK                     ;
                                    07             67   E8 00035    BLBS    UAF$GL_CTLMSK, 1$                             ; 1640
                                    51             01   7D 00038    MOVQ    #1, DEF_SYS                                   ; 1644
                                                   50   D4 0003B    CLRL    FLAG                                          ; 1645
                                                   06   11 0003D    BRB     2$                                            ; 1640
                                    52             01   D0 0003F 1$: MOVL    #1, LOCK_REC                                  ; 1649
                                    50             01   7D 00042    MOVQ    #1, FLAG                                      ; 1651
                                                   50   DD 00045 2$: PUSHL   FLAG                                         ; 1662
                                                   51   DD 00047    PUSHL   DEF_SYS                                       ;
                                                   52   DD 00049    PUSHL   LOCK_REC                                      ;
                    00000000V   00                 03   FB 0004B    CALLS   #3, GET_USER_RECORD                           ;
```

```
                    03              50  E8 00052          BLBS    R0, 3$                                  ;
                                  0169  31 00055          BRW     25$                                     ;
                                    59  DD 00058 3$:      PUSHL   R9                                      ; 1665
                                    6B  01  FB 0005A       CALLS   #1, CLI$PRESENT                         ;
                                    14  50  E9 0005D       BLBC    R0, 4$                                  ;
                                 5C     A7  9F 00060       PUSHAB  TOKENDSC                                ; 1666
                                    59  DD 00063          PUSHL   R9                                      ;
                00000000G           00  02  FB 00065     CALLS   #2, CLI$GET_VALUE                       ;
                                    05  50  E9 0006C       BLBC    R0, 4$                                  ;
                                 5C     A7  B5 0006F       TSTW    TOKENLEN                                ; 1667
                                    08  12  00072         BNEQ    5$                                      ;
                             00000000G  8F  DD 00074 4$:    PUSHL   #UAF$_NOUSERNAME                        ; 1668
                                    37  11  0007A         BRB     9$                                      ;
                                    0C  5C  A7  B1 0007C 5$:    CMPW    TOKENLEN, #12                   ; 1674
                                    09  1B  00080         BLEQU   6$                                      ;
                             00000000G  8F  DD 00082       PUSHL   #UAF$_NAMETOOBIG                        ; 1675
                                    68  01  FB 00088       CALLS   #1, LIB$SIGNAL                          ;
                                    53  5C  A7  3C 0008B 6$:    MOVZWL  TOKENLEN, R3                    ; 1680
                                    53  D7  0008F         DECL    R3                                      ;
                                    52  D4  00091         CLRL    I                                       ;
                                    22  11  00093         BRB     11$                                     ;
                                    51  01DC  C9  9A 00095 7$:    MOVZBL  SYMBOL_STR, R1                  ; 1682
                                    50  60  A7  D0 0009A       MOVL    TOKENPTR, R0                        ; 1684
01DD  C9                            51      4240  3A 0009E     LOCC    (I)[R0], R1, SYMBOL_STR+1               ;
                                    02  12  000A5         BNEQ    8$                                      ;
                                    51  D4  000A7         CLRL    R1                                      ;
                                    51  D5  000A9 8$:      TSTL    R1                                      ;
                                    08  12  000AB         BNEQ    10$                                     ;
                             00000000G  8F  DD 000AD       PUSHL   #UAF$_INVUSERNAME                       ; 1685
                                    78  11  000B3 9$:      BRB     14$                                     ;
                                    52  D6  000B5 10$:     INCL    I                                       ; 1684
                                53      52  D1  000B7 11$:     CMPL    I, R3                               ;
                                    D9  1B  000BA         BLEQU   7$                                      ;
                                    10  67  E9  000BC     BLBC    UAF$GL_CTLMSK, 12$                      ; 1690
                                    56  5C  A7  3C 000BF     MOVZWL  TOKENLEN, R6                        ; 1693
                                    50  60  A7  D0 000C3     MOVL    TOKENPTR, R0                            ;
                             6A     60  56  28  000C7       MOVC3   R6, (R0), NEWUSERNAME                   ;
                                FC  AA  56  D0  000CB       MOVL    R6, NEWUSERLEN                          ; 1694
                             6E 0080  C7  20  28 000CF 12$:     MOVC3   #32, RECBUF+4, OLD_USER_BUFFER  ; 1700
                                    50  60  A7  D0 000D5     MOVL    TOKENPTR, R0                            ; 1701
                          20 20     60  5C  A7  2C 000D9     MOVC5   TOKENLEN, (R0), #32, #32, RECBUF+4      ; 1702
                                                0080  C7      000DF
                                06F4  C7  01  D0  000E2     MOVL    #1, PWD_FLAG                            ; 1703
                             00000000G  00  FB  000E7       CALLS   #0, UPDATE_RECORD                       ; 1705
                                    4D  50  E9  000EE       BLBC    STATUS, 16$                             ; 1704
                                    14  67  E8  000F1       BLBS    UAF$GL_CTLMSK, 13$                      ; 1712
                                            01E0  C7  B4 000F4  CLRW    RECBUF+356                          ; 1715
                          08 00         6E  00  2C  000F8     MOVC5   #0, (SP), #0, #8, RECBUF+396            ; 1716
                                            0208  C7      000FD
                          08 00         6E  00  2C  00100     MOVC5   #0, (SP), #0, #8, RECBUF+404            ; 1717
                                            0210  C7      00105
                                            0664  C7  9F  00108 13$:    PUSHAB  UAFRAB                      ; 1723
                             00000000G  00  01  FB  0010C     CALLS   #1, SYS$PUT                             ;
                                   74   A7  50  D0  00113     MOVL    R0, RMSERR                              ;
                                    27  50  E8  00117       BLBS    R0, 17$                                 ;
                                    50  74  A7  D0 0011A       MOVL    RMSERR, R0                          ; 1726
                             000184EC   8F  50  D1  0011E     CMPL    R0, #99564                              ;
```

```
                                    0A  12 00125           BNEQ    15$                                     ;
                        00000000G   8F  DD 00127           PUSHL   #UAF$_UAEERR                            ; 1728
                                68  01  FB 00129 14$:      CALLS   #1, LIB$SIGNAL                          ;
                                    04  00130           RET                                             ; 1730
                                50  DD  00131 15$:      PUSHL   R0                                      ; 1731
                                7E  D4  00133           CLRL    -(SP)                                   ;
                        00000000G   8F  DD 00135           PUSHL   #UAF$_ADDERR                            ;
                                68  03  FB 00138           CALLS   #3, LIB$SIGNAL                          ;
                              0080  31  00013E 16$:      BRW     25$                                     ;
                            CC  AA      01  D0 00141 17$:      MOVL    #1, MODIFY_FLAG                         ; 1732
                                5E  DD  00145           PUSHL   SP                                      ; 1740
                                08  67  E8 00147           BLBS    UAF$GL_CTLMSK, 18$                      ; 1741
                         00040002   8F  DD 0014A           PUSHL   #262146                                 ;
                                    06  11 00150           BRB     19$                                     ;
                         00050002   8F  DD 00152 18$:      PUSHL   #327682                                 ;
                    00000000V   00      02  FB 00158 19$:      CALLS   #2, SECURITY_AUDIT                      ;
                                5B  67  E8 0015F           BLBS    UAF$GL_CTLMSK, 24$                      ; 1745
                        00000000G   8F  DD 00162           PUSHL   #UAF$_COPMSG                            ; 1748
                                68  01  FB 00168           CALLS   #1, LIB$SIGNAL                          ;
                            11  67      03  E1 0016B           BBC     #3, UAF$GL_CTLMSK, 20$                  ; 1749
                            0D  67      04  E0 0016F           BBS     #4, UAF$GL_CTLMSK, 20$                  ; 1750
                            09  67      05  E1 00173           BBC     #5, UAF$GL_CTLMSK, 20$                  ; 1751
                        00000000G   8F  DD 00177           PUSHL   #UAF$_CLIWARNMSG                        ; 1752
                                68  01  FB 0017D           CALLS   #1, LIB$SIGNAL                          ;
                                09  06F4   C7  E9 00180 20$:      BLBC    PWD_FLAG, 21$                   ; 1757
                        00000000G   8F  DD 00185           PUSHL   #UAF$_DEFPWD                            ; 1758
                                68  01  FB 0018B           CALLS   #1, LIB$SIGNAL                          ;
                                67  04  8A 0018E 21$:      BICB2   #4, UAF$GL_CTLMSK                       ; 1759
                                  34    A9  9F 00191           PUSHAB  SD_ADD_IDENTIFIER                       ; 1760
                                6B  01  FB 00194           CALLS   #1, CLI$PRESENT                         ;
                                23  50  E9 00197           BLBC    R0, 24$                                 ;
                                1F      70  A7  E9 0019A           BLBC    RDB_EXISTS, 24$                 ; 1761
                                  80    AA  9F 0019E           PUSHAB  SD_ATTRIBRESOURCE                       ; 1768
                                6B  01  FB 001A1           CALLS   #1, CLI$PRESENT                         ;
                                09  50  E9 001A4           BLBC    R0, 22$                                 ;
                    00000000G   00      01  D0 001A7           MOVL    #1, ATTRIBUTES                          ; 1769
                                    06  11 001AE           BRB     23$                                     ;
                         00000000G  00  D4  001B0 22$:      CLRL    ATTRIBUTES                              ; 1770
                    00000000G   00      00  FB 001B6 23$:      CALLS   #0, UAF$ADD_IDENT_RECBUF                ; 1771
                                50  01  D0 001BD 24$:      MOVL    #1, R0                                  ; 1785
                                    04  001C0           RET                                             ;
                                50  D4  001C1 25$:      CLRL    R0                                      ; 1787
                                    04  001C3           RET                                             ;

; Routine Size:  452 bytes,    Routine Base:  $CODE$ + 06CB
```

```
;  1707          1788  1 %sbttl 'create_proxy - create NETUAF.DAT proxy file'
;  1708          1789  1 global routine create_proxy : novalue =
;  1709          1790  2 begin
;  1710          1791  2
;  1711          1792  2 !++
;  1712          1793  2 !
;  1713          1794  2 !
;  1714          1795  2 ! FUNCTIONAL DESCRIPTION:
;  1715          1796  2 !
;  1716          1797  2 !       This routine will create a DECnet Proxy Login File,
;  1717          1798  2 !       if and only if it does not already exist,
;  1718          1799  2 !       called NETUAF.DAT, in order to map remote users into
;  1719          1800  2 !       local accounts.
;  1720          1801  2 !
;  1721          1802  2 ! INPUTS:
;  1722          1803  2 !
;  1723          1804  2 !    none
;  1724          1805  2 !
;  1725          1806  2 ! OUTPUTS:
;  1726          1807  2 !
;  1727          1808  2 !    none
;  1728          1809  2 !
;  1729          1810  2 ! IMPLICIT INPUTS:
;  1730          1811  2 !
;  1731          1812  2 !       none
;  1732          1813  2 !
;  1733          1814  2 ! IMPLICIT OUTPUTS:
;  1734          1815  2 !
;  1735          1816  2 !       none
;  1736          1817  2 !
;  1737          1818  2 ! SIDE EFFECTS:
;  1738          1819  2 !
;  1739          1820  2 !       NETUAF.DAT is created and initialized
;  1740          1821  2 !
;  1741          1822  2 !--
;  1742          1823  2
;  1743          1824  2 !
;  1744          1825  2 ! NETUAF.DAT should not already exist
;  1745          1826  2 !
;  1746          1827  2 if .netuaf_exists
;  1747          1828  2 then
;  1748          1829  3     begin
;  1749          1830  3     LIB$SIGNAL(UAF$_NAFAEX);
;  1750          1831  3     return;
;  1751          1832  3     end;
;  1752          1833  2
;  1753          1834  2 !
;  1754          1835  2 ! Should be able to create NETUAF.DAT with no problems
;  1755          1836  2 !
;  1756          1837  3 if rmsbad ($create (fab = naffab))
;  1757          1838  2 then LIB$SIGNAL(UAF$_NAFCREERR, 0, .rmserr);
;  1758          1839  2
;  1759          1840  2 !
;  1760          1841  2 ! Should connect ok, too
;  1761          1842  2 !
;  1762          1843  3 if rmsbad ($connect (rab = nafrab))
;  1763          1844  2 then LIB$SIGNAL(UAF$_NAFCONERR, 0, .rmserr);
```

```
;  1764          1845  2
;  1765          1846  2 !
;  1766          1847  2 ! Normal access is by key
;  1767          1848  2 !
;  1768          1849  2 nafrab [rab$b_rac] = rab$c_key;
;  1769          1850  2
;  1770          1851  2 !
;  1771          1852  2 ! Establish proper addresses
;  1772          1853  2 !
;  1773          1854  2 nafrab [rab$l_ubf] = netbuf;
;  1774          1855  2 nafrab [rab$l_rbf] = netbuf;
;  1775          1856  2
;  1776          1857  2 !
;  1777          1858  2 ! Set NETUAF.DAT existence flag
;  1778          1859  2 !
;  1779          1860  2 netuaf_exists = true;
;  1780          1861  2 netuaf_modified = true;
;  1781          1862  2
;  1782          1863  1 end;

                                 000C 00000          .ENTRY  CREATE_PROXY, Save R2,R3                ;  1789
                           53 00000000G   00  9E 00002  MOVAB   LIB$SIGNAL, R3                      ;
                           52 00000000'   00  9E 00009  MOVAB   RMSERR, R2                          ;
                           0A         F8  A2  E9 00010  BLBC    NETUAF_EXISTS, 1$                   ;  1827
                              00000000G   8F  DD 00014  PUSHL   #UAF$_NAFAEX                        ;  1830
                           63             01  FB 0001A  CALLS   #1, LIB$SIGNAL                     ;
                                          04  0001D  RET                                           ;  1829
                              00000000'   00  9F 0001E  1$:     PUSHAB  NAFFAB                      ;  1837
                     00000000G   00       01  FB 00024  CALLS   #1, SYS$CREATE                     ;
                                 62       50  D0 0002B  MOVL    R0, RMSERR                          ;
                                 0D       50  E8 0002E  BLBS    R0, 2$                              ;
                                          62  DD 00031  PUSHL   RMSERR                            ;  1838
                                          7E  D4 00033  CLRL    -(SP)                               ;
                                 00000000G   8F  DD 00035  PUSHL   #UAF$_NAFCREERR                  ;
                                 63          03  FB 0003B  CALLS   #3, LIB$SIGNAL                   ;
                                       0634  C2  9F 0003E  2$:     PUSHAB  NAFRAB                  ;  1843
                     00000000G   00       01  FB 00042  CALLS   #1, SYS$CONNECT                    ;
                                 62       50  D0 00049  MOVL    R0, RMSERR                          ;
                                 0D       50  E8 0004C  BLBS    R0, 3$                              ;
                                          62  DD 0004F  PUSHL   RMSERR                            ;  1844
                                          7E  D4 00051  CLRL    -(SP)                               ;
                                 00000000G   8F  DD 00053  PUSHL   #UAF$_NAFCONERR                  ;
                                 63          03  FB 00059  CALLS   #3, LIB$SIGNAL                   ;
                              0652       C2       01  90 0005C  3$:     MOVB    #1, NAFRAB+30       ;  1849
                              0658       C2  058C  C2  9E 00061  MOVAB   NETBUF, NAFRAB+36           ;  1854
                              065C       C2  058C  C2  9E 00068  MOVAB   NETBUF, NAFRAB+40           ;  1855
                                F8       A2       01  D0 0006F  MOVL    #1, NETUAF_EXISTS           ;  1860
                           00000000'      00       01  D0 00073  MOVL    #1, NETUAF_MODIFIED         ;  1861
                                                    04  0007A  RET                                   ;  1863

; Routine Size:  123 bytes,    Routine Base:  $CODE$ + 088F
```

```
;  1784          1864  1 %sbttl 'modify_uaf - update user record (s)'
;  1785          1865  1 global routine modify_uaf : novalue =
;  1786          1866  2 begin
;  1787          1867  2
;  1788          1868  2 !++
;  1789          1869  2 !
;  1790          1870  2 ! FUNCTIONAL DESCRIPTION:
;  1791          1871  2 !
;  1792          1872  2 !       Routine to modify any of the fields in one or more user records.
;  1793          1873  2 !
;  1794          1874  2 ! INPUTS:
;  1795          1875  2 !
;  1796          1876  2 !       none
;  1797          1877  2 !
;  1798          1878  2 ! IMPLICIT INPUTS:
;  1799          1879  2 !
;  1800          1880  2 !       none
;  1801          1881  2 !
;  1802          1882  2 ! OUTPUTS:
;  1803          1883  2 !
;  1804          1884  2 !       none
;  1805          1885  2 !
;  1806          1886  2 ! IMPLICIT OUTPUTS:
;  1807          1887  2 !
;  1808          1888  2 !       none
;  1809          1889  2 !
;  1810          1890  2 ! SIDE EFFECTS:
;  1811          1891  2 !
;  1812          1892  2 !       none
;  1813          1893  2 !
;  1814          1894  2 ! ROUTINE VALUE:
;  1815          1895  2 !
;  1816          1896  2 !       none
;  1817          1897  2 !--
;  1818          1898  2
;  1819          1899  2 local
;  1820          1900  2         status;
;  1821          1901  2
;  1822          1902  2 !
;  1823          1903  2 ! Obtain the user specification.  This sets wildcard flags and initializes
;  1824          1904  2 ! the appropriate key in RECBUF.
;  1825          1905  2 !
;  1826          1906  2
;  1827          1907  2 if not parse_wild (sd_token1,false)              ! Null string is disallowed
;  1828          1908  2 then return;
;  1829          1909  2
;  1830          1910  2 uafrab[rab$l_rop] = rab$m_rlk;           ! Lock records for writing
;  1831          1911  2 rabptr = outrab;
;  1832          1912  2 found_match = false;
;  1833          1913  2
;  1834          1914  3 if rmsbad (status = wild_user (modify_rec))      ! Modify each record
;  1835          1915  2 then
;  1836          1916  3     begin
;  1837          1917  3     if .rmserr eql rms$_rnf
;  1838          1918  3     then
;  1839          1919  3         LIB$SIGNAL(UAF$_BADSPC)
;  1840          1920  3     else
```

```
;  1841          1921  4          (if .rmserr neq 0
;  1842          1922  4           then
;  1843          1923  4               LIB$SIGNAL(UAF$_MDFYERR, 0, .rmserr))
;  1844          1924  3      end
;  1845          1925  2  else
;  1846          1926  3      begin
;  1847          1927  3      if .status and .found_match
;  1848          1928  3      then
;  1849          1929  4          begin
;  1850          1930  4          LIB$SIGNAL(UAF$_MDFYMSG);
;  1851          1931  5          if (.uaf$gl_ctlmsk[uaf$v_cli]
;  1852          1932  5          and not .uaf$gl_ctlmsk[uaf$v_clitables])
;  1853          1933  4          and .uaf$gl_ctlmsk[uaf$v_clitab_pres]
;  1854          1934  4          then LIB$SIGNAL(UAF$_CLIWARNMSG);
;  1855          1935  3          end;
;  1856          1936  2      end;
;  1857          1937  1  end;


                                                001C 00000     .ENTRY  MODIFY_UAF, Save R2,R3,R4                       ; 1865
                                    54 00000000G   00   9E 00002 MOVAB   LIB$SIGNAL, R4                                 ;
                                    53 00000000'   00   9E 00009 MOVAB   UAF$GL_CTLMSK, R3                              ;
                                                   7E   D4 00010 CLRL    -(SP)                                         ; 1907
                                       00000000'   00   9F 00012 PUSHAB  SD_TOKEN1                                     ;
                                00000000G  00      02   FB 00018 CALLS   #2, PARSE_WILD                               ;
                                           71      50   E9 0001F BLBC    R0, 4$                                        ;
                                     0668  C3 00080000   8F   D0 00022 MOVL    #524288, UAFRAB+4                     ; 1910
                                       F4  A3      06F8   C3   9E 0002B MOVAB   OUTRAB, RABPTR                        ; 1911
                                           A3      073C   C3   D4 00031 CLRL    FOUND_MATCH                           ; 1912
                                       00000000V   00   9F 00035 PUSHAB  MODIFY_REC                                    ; 1914
                                00000000V  00      01   FB 0003B CALLS   #1, WILD_USER                              ;
                                       74  A3              50   D0 00042 MOVL    STATUS, RMSERR                        ;
                                           27              50   E8 00046 BLBS    STATUS, 2$                            ;
                                           52          74  A3   D0 00049 MOVL    RMSERR, R2                            ; 1917
                                00182B2  8F                52   D1 0004D CMPL    R2, #98994                            ;
                                                           08   12 00054 BNEQ    1$                                    ;
                                           00000000G       8F   DD 00056 PUSHL   #UAF$_BADSPC                          ; 1919
                                                           32   11 0005C BRB     3$                                    ;
                                                           52   D5 0005E 1$:    TSTL    R2                            ; 1921
                                                           31   13 00060 BEQL    4$                                    ;
                                                           52   DD 00062 PUSHL   R2                                    ; 1923
                                                           7E   D4 00064 CLRL    -(SP)                                 ;
                                           00000000G       8F   DD 00066 PUSHL   #UAF$_MDFYERR                         ;
                                       64                  03   FB 0006C CALLS   #3, LIB$SIGNAL                       ;
                                                           04   00 0006F RET                                          ; 1916
                                       1E      073C   C3   E9 00070 2$:    BLBC    FOUND_MATCH, 4$                   ; 1927
                                           00000000G       8F   DD 00075 PUSHL   #UAF$_MDFYMSG                         ; 1930
                                       64                  01   FB 0007B CALLS   #1, LIB$SIGNAL                       ;
                                11     63                  03   E1 0007E BBC     #3, UAF$GL_CTLMSK, 4$              ; 1931
                                0D     63                  04   E0 00082 BBS     #4, UAF$GL_CTLMSK, 4$              ; 1932
                                09     63                  05   E1 00086 BBC     #5, UAF$GL_CTLMSK, 4$              ; 1933
                                           00000000G       8F   DD 0008A PUSHL   #UAF$_CLIWARNMSG                      ; 1934
                                       64                  01   FB 00090 3$:    CALLS   #1, LIB$SIGNAL                ;
                                                           04   00 00093 4$:    RET                                   ; 1937
```

; Routine Size: 148 bytes,   Routine Base: $CODE$ + 090A

```
;  1859          1938  1 %sbttl 'modify_rec - update a user record action routine'
;  1860          1939  1 routine modify_rec =
;  1861          1940  2 begin
;  1862          1941  2
;  1863          1942  2 !++
;  1864          1943  2 !
;  1865          1944  2 ! FUNCTIONAL DESCRIPTION:
;  1866          1945  2 !
;  1867          1946  2 !       Modify an individual user record.
;  1868          1947  2 !
;  1869          1948  2 ! INPUTS:
;  1870          1949  2 !
;  1871          1950  2 !       none
;  1872          1951  2 !
;  1873          1952  2 ! IMPLICIT INPUTS:
;  1874          1953  2 !
;  1875          1954  2 !       RABPTR - RMS data structure for the file
;  1876          1955  2 !
;  1877          1956  2 ! OUTPUTS:
;  1878          1957  2 !
;  1879          1958  2 !       none
;  1880          1959  2 !
;  1881          1960  2 ! IMPLICIT OUTPUTS:
;  1882          1961  2 !
;  1883          1962  2 !       none
;  1884          1963  2 !
;  1885          1964  2 ! SIDE EFFECTS:
;  1886          1965  2 !
;  1887          1966  2 !       none
;  1888          1967  2 !
;  1889          1968  2 ! ROUTINE VALUE:
;  1890          1969  2 !
;  1891          1970  2 !       If an error is encountered the appropriate status is returned
;  1892          1971  2 !       except if the error message has already been output in which
;  1893          1972  2 !       case 0 is returned.
;  1894          1973  2 !--
;  1895          1974  2
;  1896          1975  2 local
;  1897          1976  2     old_uic     : $bblock[4],
;  1898          1977  2     new_uic     : $bblock[4],
;  1899          1978  2     oldaccname  : vector [uaf$s_account,byte],
;  1900          1979  2     newaccname  : vector [uaf$s_account,byte],
;  1901          1980  2     oldaccdesc  : statdesc ,
;  1902          1981  2     newaccdesc  : statdesc ,
;  1903          1982  2     status      : long ;
;  1904          1983  2
;  1905          1984  2 !
;  1906          1985  2 ! User record has been read into RECBUF by caller.  Update values
;  1907          1986  2 ! and modify the record.
;  1908          1987  2 !
;  1909          1988  2 ! When accessing records by uic, this routine is called repeatedly
;  1910          1989  2 ! from WILD_USER. UPDATE_RECORD is called to modify the appropriate
;  1911          1990  2 ! record fields for each requested record, and therefore must
;  1912          1991  2 ! reparse the command line each time.  If call_count is greater
;  1913          1992  2 ! than 0, the command line is reparsed.
;  1914          1993  2 !
;  1915          1994  2
```

```
;  1916          1995  2 !IF .by_account
;  1917          1996  2 !THEN
;  1918          1997  2 !    (IF NOT fmg$match_name (NAMELEN (UAF$S_ACCOUNT,UAF$T_ACCOUNT),
;  1919          1998  2 !                                    RECBUF[UAF$T_ACCOUNT],
;  1920          1999  2 !                                   .match_tokenlen, match_token)
;  1921          2000  2 !     THEN
;  1922          2001  2 !       RETURN TRUE)
;  1923          2002  2 !ELSE
;  1924          2003  2 if .str_wild
;  1925          2004  2 and not fmg$match_name (namelen (uaf$s_username,recbuf[uaf$t_username]),
;  1926          2005  2                         recbuf[uaf$t_username],
;  1927          2006  2                         .match_tokenlen,  match_token)
;  1928          2007  2 then return true;
;  1929          2008  2 found_match = true;
;  1930          2009  2
;  1931          2010  2 if ch$eql (.defuser<0,8>, defuser+1, .tokenlen, .tokenptr, ' ')
;  1932          2011  2 or ch$eql (.defuser<0,8>, defuser+1,
;  1933          2012  2           namelen (uaf$s_username, recbuf[uaf$t_username]),
;  1934          2013  2           recbuf[uaf$t_username], ' ')
;  1935          2014  2 then
;  1936          2015  3     begin
;  1937          2016  3     mod_default = true;
;  1938          2017  3     default_uaf ();
;  1939          2018  3     call_count = .call_count + 1;
;  1940          2019  3     return true;
;  1941          2020  3     end;
;  1942          2021  2
;  1943          2022  2 !
;  1944          2023  2 ! Save the old UIC and account name
;  1945          2024  2 !
;  1946          2025  2 old_uic[uic$v_format] = 0 ;
;  1947          2026  2 old_uic[uic$v_group ] = .recbuf [uaf$w_grp] ;
;  1948          2027  2 old_uic[uic$v_member] = .recbuf [uaf$w_mem] ;
;  1949          2028  2 ch$move ( uaf$s_account, recbuf[uaf$t_account], oldaccname ) ;
;  1950          2029  2 oldaccdesc [length]  = namelen ( uaf$s_account, recbuf[uaf$t_account]) ;
;  1951          2030  2 oldaccdesc [pointer] = oldaccname ;
;  1952          2031  2
;  1953          2032  2 if update_record ()
;  1954          2033  2 then
;  1955          2034  3     begin
;  1956          2035  3     !
;  1957          2036  3     ! Save the new UIC and account name
;  1958          2037  3     !
;  1959          2038  3     new_uic[uic$v_format] = 0 ;
;  1960          2039  3     new_uic[uic$v_group ] = .recbuf [uaf$w_grp] ;
;  1961          2040  3     new_uic[uic$v_member] = .recbuf [uaf$w_mem] ;
;  1962          2041  3     ch$move ( uaf$s_account, recbuf[uaf$t_account], newaccname ) ;
;  1963          2042  3     newaccdesc [length]  = namelen ( uaf$s_account, recbuf[uaf$t_account]) ;
;  1964          2043  3     newaccdesc [pointer] = newaccname ;
;  1965          2044  3
;  1966          2045  3     !
;  1967          2046  3     ! Update the UAF record
;  1968          2047  3     !
;  1969          2048  4     if rmsbad ($update (rab = uafrab))
;  1970          2049  3     then
;  1971          2050  4         begin
;  1972          2051  4         LIB$SIGNAL(UAF$_MDFYERR, 0, .rmserr);
```

```
;  1973          2052  4          return .rmserr
;  1974          2053  4          end
;  1975          2054  3      else
;  1976          2055  4          begin
;  1977          2056  4          modify_flag = true;
;  1978          2057  4          security_audit (nsa$k_recid_sysuaf_mod);
;  1979          2058  4          call_count = .call_count + 1;
;  1980          2059  3          end;
;  1981          2060  3
;  1982          2061  4      if (cli$present ( sd_modify_identifier ) and
;  1983          2062  4          .rdb_exists )
;  1984          2063  3      then
;  1985          2064  4          begin
;  1986          2065  4
;  1987          2066  4          !
;  1988          2067  4          ! If the UIC changed then modify the identifiers
;  1989          2068  4          !
;  1990          2069  4          if .old_uic neq .new_uic
;  1991          2070  4          then
;  1992          2071  5              begin
;  1993          2072  5              local
;  1994          2073  5                  oldidname       : vector [kgb$s_name, byte],
;  1995          2074  5                  oldiddesc       : statdesc ;
;  1996          2075  5
;  1997          2076  5              oldiddesc[length]  = kgb$s_name ;
;  1998          2077  5              oldiddesc[pointer] = oldidname ;
;  1999        P 2078  5              status = $idtoasc ( id     = .old_uic,
;  2000        P 2079  5                                  namlen = oldiddesc[length],
;  2001          2080  5                                  nambuf = oldiddesc ) ;
;  2002          2081  5
;  2003          2082  5              if not .status
;  2004          2083  5              then LIB$SIGNAL(UAF$_RDBMDFYERRU, 2,
;  2005          2084  5                                  .old_uic[uic$v_group],
;  2006          2085  5                                  .old_uic[uic$v_member], .status)
;  2007          2086  5              else if .status
;  2008          2087  5                  then
;  2009          2088  6                      begin
;  2010        P 2089  6                      status = $mod_ident ( id        = .old_uic,
;  2011          2090  6                                            new_value = .new_uic ) ;
;  2012          2091  6
;  2013          2092  6                      if not .status
;  2014          2093  6                          then LIB$SIGNAL(UAF$_RDBMDFYERR, 1, oldiddesc, .status)
;  2015          2094  6                          else
;  2016          2095  7                              begin
;  2017          2096  7                              rightslist_modified = true ;
;  2018          2097  7                              LIB$SIGNAL(UAF$_RDBMDFYMSG, 1, oldiddesc);
;  2019          2098  6                              end ;
;  2020          2099  5                      end ;
;  2021          2100  4              end;
;  2022          2101  3          end ;
;  2023          2102  3      return true ;
;  2024          2103  3      end
;  2025          2104  2  else
;  2026          2105  3      begin
;  2027          2106  3      $release (rab = uafrab);
;  2028          2107  3      return false
;  2029          2108  3      end
```

```
: 2030          2109  1 end;


                                                        .EXTRN  SYS$UPDATE, SYS$MOD_IDENT
                                                        .EXTRN  SYS$RELEASE

                                        07FC 00000 MODIFY_REC:
                                                        .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10                ; 1939
                                5A 00000000G    00  9E 00002         MOVAB   LIB$SIGNAL, R10
                                59 00000000'    00  9E 00009         MOVAB   DEFUSER+1, R9
                                58 00000000'    00  9E 00010         MOVAB   RECBUF+4, R8
                                5E      88      AE  9E 00017         MOVAB   -120(SP), SP
                             30 AE 010E0000     8F  D0 0001B         MOVL    #17694720, OLDACCDESC                  ; 1980
                                       34       AE  D4 00023         CLRL    OLDACCDESC+4
                             28 AE 010E0000     8F  D0 00026         MOVL    #17694720, NEWACCDESC                  ; 1981
                                       2C       AE  D4 0002E         CLRL    NEWACCDESC+4
                                1F     06CC     C8  E9 00031         BLBC    STR_WILD, 1$                           ; 2003
                                55      8C      A8  9E 00036         MOVAB   MATCH_TOKEN, R5                        ; 2005
                                53              68  9E 0003A         MOVAB   RECBUF+4, R3
                          68                    20  3A 0003D         LOCC    #32, #32, RECBUF+4                     ; 2004
                                52      E0      A0  9E 00041         MOVAB   -32(R0), R2
                                52              52  CE 00045         MNEGL   R2, R2
                                54      D0      A8  D0 00048         MOVL    MATCH_TOKENLEN, R4                     ; 2005
                                   00000000G    00  16 0004C         JSB     FMG$MATCH_NAME
                                37              50  E9 00052         BLBC    R0, 3$
                             06BC   C8          01  D0 00055 1$:    MOVL    #1, FOUND_MATCH                        ; 2008
                                54      FF      A9  9A 0005A         MOVZBL  DEFUSER, R4                            ; 2010
                                50      E0      A8  D0 0005E         MOVL    TOKENPTR, R0
                   DC   A8              20      69  54 2D 00062      CMPC5   R4, DEFUSER+1, #32, TOKENLEN, (R0)
                                                    60 00068
                                                    10 13 00069      BEQL    2$
                                   68               20  3A 0006B     LOCC    #32, #32, RECBUF+4                     ; 2012
                                   50               20  C3 0006F     SUBL3   R0, #32, R0
                             50                     20  54 2D 00073  CMPC5   R4, DEFUSER+1, #32, R0, RECBUF+4      ; 2013
                                                    68 00078
                                                    14 12 00079      BNEQ    4$
                                   00000000'    00      01  D0 0007B 2$:    MOVL    #1, MOD_DEFAULT                ; 2016
                                   00000000V    00      00  FB 00082      CALLS   #0, DEFAULT_UAF                ; 2017
                                                    D8  A8  D6 00089     INCL    CALL_COUNT                     ; 2018
                                                  0122  31  0008C 3$:    BRW     10$                            ; 2019
                             56                     02  F0 0008F 4$:    INSV    #0, #30, #2, OLD_UIC           ; 2025
                             56                     0E  22  A8 F0 00094  INSV    RECBUF+38, #16, #14, OLD_UIC   ; 2026
                                                        20  A8 B0 0009A  MOVW    RECBUF+36, OLD_UIC             ; 2027
                                   58   AE      30      20  28 0009E     MOVC3   #32, RECBUF+52, OLDACCNAME     ; 2028
                                   30   A8              20  3A 000A4     LOCC    #32, #32, RECBUF+52            ; 2029
                                   30   AE              50  A3 000A9     SUBW3   R0, #32, OLDACCDESC
                                              34       AE  58  AE 9E 000AE  MOVAB   OLDACCNAME, OLDACCDESC+4      ; 2030
                                   00000000G    00      00  FB 000B3     CALLS   #0, UPDATE_RECORD              ; 2032
                                                        50  E8 000BA     BLBS    R0, 5$
                                                      00F5  31  000BD     BRW     11$
                             57                     02  F0 000C0 5$:    INSV    #0, #30, #2, NEW_UIC           ; 2038
                             57                     0E  22  A8 F0 000C5  INSV    RECBUF+38, #16, #14, NEW_UIC   ; 2039
                                                        20  A8 B0 000CB  MOVW    RECBUF+36, NEW_UIC             ; 2040
                                   38   AE      30      20  28 000CF     MOVC3   #32, RECBUF+52, NEWACCNAME     ; 2041
                                   30   A8              20  3A 000D5     LOCC    #32, #32, RECBUF+52            ; 2042
                                   28   AE              20  50 A3 000DA  SUBW3   R0, #32, NEWACCDESC
```

```
                 2C   AE        38   AE  9E 000DF          MOVAB   NEWACCNAME, NEWACCDESC+4                ; 2043
                            05E4   C8  9F 000E4          PUSHAB  UAFRAB                                  ; 2048
          00000000G  00           01  FB 000E8          CALLS   #1, SYS$UPDATE
                 F4  A8           50  D0 000EF          MOVL    R0, RMSERR
                     13           50  E8 000F3          BLBS    R0, 6$
                                 F4  A8  DD 000F6          PUSHL   RMSERR                                  ; 2051
                                     7E  D4 000F9          CLRL    -(SP)
                          00000000G  8F  DD 000FB          PUSHL   #UAF$_MDFYERR
                     6A           03  FB 00101          CALLS   #3, LIB$SIGNAL
                     50          F4   A8  D0 00104          MOVL    RMSERR, R0                              ; 2052
                                      04  00108          RET
          00000000'  00           01  D0 00109 6$:      MOVL    #1, MODIFY_FLAG                         ; 2056
                          00030002   8F  DD 00110          PUSHL   #196610                                 ; 2057
          00000000V  00           01  FB 00116          CALLS   #1, SECURITY_AUDIT
                              D8  A8  D6 0011D          INCL    CALL_COUNT                              ; 2058
                            FF63  C9  9F 00120          PUSHAB  SD_MODIFY_IDENTIFIER                    ; 2061
          00000000G  00           01  FB 00124          CALLS   #1, CLI$PRESENT
                     70           50  E9 0012B          BLBC    R0, 8$
                     7F          F0   A8  E9 0012E          BLBC    RDB_EXISTS, 10$                         ; 2062
                     57           56  D1 00132          CMPL    OLD_UIC, NEW_UIC                        ; 2069
                                  7A  13 00135          BEQL    10$
                     6E 010E0000   8F  D0 00137          MOVL    #17694720, OLDIDDESC                    ; 2074
                            04    AE  D4 0013E          CLRL    OLDIDDESC+4
                     6E           20  B0 00141          MOVW    #32, OLDIDDESC                          ; 2076
                 04  AE        08   AE  9E 00144          MOVAB   OLDIDNAME, OLDIDDESC+4                  ; 2077
                                 7E  7C 00149          CLRQ    -(SP)                                   ; 2080
                                 7E  D4 0014B          CLRL    -(SP)
                                 0C  AE  9F 0014D          PUSHAB  OLDIDDESC
                                 10  AE  9F 00150          PUSHAB  OLDIDDESC
                                     56  DD 00153          PUSHL   OLD_UIC
                    00000000G    00      06  FB 00155          CALLS   #6, SYS$IDTOASC
                                 52          50  D0 0015C          MOVL    R0, STATUS
                                 17          52  E8 0015F          BLBS    STATUS, 7$                      ; 2082
                                             52  DD 00162          PUSHL   STATUS                          ; 2085
                                 7E          56  3C 00164          MOVZWL  OLD_UIC, -(SP)
          7E          56         0E          10  EF 00167          EXTZV   #16, #14, OLD_UIC, -(SP)        ; 2084
                                             02  DD 0016C          PUSHL   #2                              ; 2083
                          00000000G          8F  DD 0016E          PUSHL   #UAF$_RDBMDFYERRU
                                 6A          05  FB 00174          CALLS   #5, LIB$SIGNAL
                                             38  11 00177          BRB     10$
                                             57  DD 00179 7$:      PUSHL   NEW_UIC                         ; 2090
                                             7E  7C 0017B          CLRQ    -(SP)
                                             7E  D4 0017D          CLRL    -(SP)
                                             56  DD 0017F          PUSHL   OLD_UIC
                    00000000G    00      05  FB 00181          CALLS   #5, SYS$MOD_IDENT
                                 52          50  D0 00188          MOVL    R0, STATUS
                                 12          52  E8 0018B          BLBS    STATUS, 9$                      ; 2092
                                             52  DD 0018E          PUSHL   STATUS                          ; 2093
                                     04      AE  9F 00190          PUSHAB  OLDIDDESC
                                             01  DD 00193          PUSHL   #1
                          00000000G          8F  DD 00195          PUSHL   #UAF$_RDBMDFYERR
                                 6A          04  FB 0019B          CALLS   #4, LIB$SIGNAL
                                             11  11 0019E 8$:      BRB     10$
                              F8  A8         01  90 001A0 9$:      MOVB    #1, RIGHTSLIST_MODIFIED         ; 2096
                                             5E  DD 001A4          PUSHL   SP                              ; 2097
                                             01  DD 001A6          PUSHL   #1
                          00000000G          8F  DD 001A8          PUSHL   #UAF$_RDBMDFYMSG
```

```
                         6A           03  FB 001AE          CALLS   #3, LIB$SIGNAL                          ;
                         50           01  D0 001B1 10$:     MOVL    #1, R0                                 ; 2105
                                      04  001B4            RET                                            ;
                              05E4    C8  9F 001B5 11$:     PUSHAB  UAFRAB                                 ; 2106
                 00000000G   00           01  FB 001B9     CALLS   #1, SYS$RELEASE                        ;
                                          50  D4 001C0     CLRL    R0                                     ; 2107
                                              04 001C2     RET                                            ; 2109

; Routine Size:  451 bytes,    Routine Base:  $CODE$ + 099E
```

```
;  2032     2110  1 %sbttl 'remove_uaf - remove username from file'
;  2033     2111  1 global routine remove_uaf : novalue =
;  2034     2112  2 begin
;  2035     2113  2
;  2036     2114  2 !++
;  2037     2115  2 !
;  2038     2116  2 ! FUNCTIONAL DESCRIPTION:
;  2039     2117  2 !
;  2040     2118  2 !       Routine to delete a user record from the UAF file.
;  2041     2119  2 !
;  2042     2120  2 ! INPUTS:
;  2043     2121  2 !
;  2044     2122  2 !       none
;  2045     2123  2 !
;  2046     2124  2 ! OUTPUTS:
;  2047     2125  2 !
;  2048     2126  2 !       none
;  2049     2127  2 !
;  2050     2128  2 ! IMPLICIT INPUTS:
;  2051     2129  2 !
;  2052     2130  2 !       rename: a logical flag which indicates whether this COPY is part
;  2053     2131  2 !               of a RENAME operation
;  2054     2132  2 !
;  2055     2133  2 ! ROUTINE VALUE:
;  2056     2134  2 !
;  2057     2135  2 !       none
;  2058     2136  2 !
;  2059     2137  2 ! SIDE EFFECTS:
;  2060     2138  2 !
;  2061     2139  2 !  This routine also deletes any NETUAF entries for the
;  2062     2140  2 !       local user which is to be removed (If NETUAF.DAT exists)
;  2063     2141  2 !
;  2064     2142  2 !--
;  2065     2143  2
;  2066     2144  2 !
;  2067     2145  2 ! Look for the record specified by the user.  Make sure
;  2068     2146  2 ! the DEFAULT and SYSTEM records are not removed.
;  2069     2147  2 !
;  2070     2148  2
;  2071     2149  2 if get_user_record (true, false)
;  2072     2150  2 then
;  2073     2151  2
;  2074     2152  2     !
;  2075     2153  2     ! User record has been found and read into RECBUF
;  2076     2154  2     ! Zero the username field and update the record in the file
;  2077     2155  2     !
;  2078     2156  3         begin
;  2079     2157  4         if rmsbad ($delete (rab=uafrab))
;  2080     2158  3         then
;  2081     2159  3             LIB$SIGNAL(UAF$_REMERR, 0, .rmserr)
;  2082     2160  3         else
;  2083     2161  4             begin
;  2084     2162  4             modify_flag = true;                         ! mark file as modified
;  2085     2163  4             if not .uaf$gl_ctlmsk[uaf$v_rename]
;  2086     2164  4             then
;  2087     2165  4                 security_audit (nsa$k_recid_sysuaf_del);
;  2088     2166  4             if not .uaf$gl_ctlmsk[uaf$v_rename]
```

```
;  2089          2167  4          then
;  2090          2168  5              begin
;  2091          2169  5              !
;  2092          2170  5              ! If there is a proxy login file, delete entries for
;  2093          2171  5              ! the user just removed from SYSUAF.DAT
;  2094          2172  5              !
;  2095          2173  5              if .netuaf_exists
;  2096          2174  5              then adjust_proxy (remove_records);
;  2097          2175  5              LIB$SIGNAL(UAF$_REMMSG);
;  2098          2176  5              !
;  2099          2177  5              ! Unless specifically requested not to
;  2100          2178  5              ! remove the corresponding identifier from the
;  2101          2179  5              ! rights data base.
;  2102          2180  5              !
;  2103          2181  6              if (cli$present ( sd_remove_identifier ) and
;  2104          2182  6                  .rdb_exists )
;  2105          2183  5              then
;  2106          2184  5                  uaf$remove_ident_recbuf () ;
;  2107          2185  4              end;
;  2108          2186  3          end;
;  2109          2187  2      end;
;  2110          2188  1 end;


                                                        .EXTRN  SYS$DELETE

                                        000C 00000      .ENTRY  REMOVE_UAF, Save R2,R3                  ;  2111
                53 00000000G   00         9E 00002      MOVAB   LIB$SIGNAL, R3
                52 00000000'   00         9E 00009      MOVAB   RMSERR, R2
                               7E         01 7D 00010    MOVQ    #1, -(SP)                              ;  2149
00000000V       00                        02 FB 00013    CALLS   #2, GET_USER_RECORD
                6C                        50 E9 0001A    BLBC    R0, 3$
                          05F0            C2 9F 0001D    PUSHAB  UAFRAB                                 ;  2157
00000000G       00                        01 FB 00021    CALLS   #1, SYS$DELETE
                62                        50 D0 00028    MOVL    R0, RMSERR
                0E                        50 E8 0002B    BLBS    R0, 1$
                                          62 DD 0002E    PUSHL   RMSERR                                 ;  2159
                                          7E D4 00030    CLRL    -(SP)
                     00000000G            8F DD 00032    PUSHL   #UAF$_REMERR
                63                        03 FB 00038    CALLS   #3, LIB$SIGNAL
                                          04 003B        RET
00000000'       00                        01 D0 0003C 1$: MOVL   #1, MODIFY_FLAG                        ;  2162
                42                   8C   A2 EB 00043    BLBS    UAF$GL_CTLMSK, 3$                      ;  2163
                     00020002            8F DD 00047    PUSHL   #131074                                ;  216
00000000V       00                        01 FB 0004D    CALLS   #1, SECURITY_AUDIT
                31                   8C   A2 EB 00054    BLBS    UAF$GL_CTLMSK, 3$                      ;  2166
                09                   F8   A2 E9 00058    BLBC    NETUAF_EXISTS, 2$                      ;  2173
                                          01 DD 0005C    PUSHL   #1                                     ;  2174
00000000V       00                        01 FB 0005E    CALLS   #1, ADJUST_PROXY
                     00000000G            8F DD 00065 2$: PUSHL  #UAF$_REMMSG                           ;  2175
                63                        01 FB 0006B    CALLS   #1, LIB$SIGNAL
                     00000000'            00 9F 0006E    PUSHAB  SD_REMOVE_IDENTIFIER                   ;  2181
00000000G       00                        01 FB 00074    CALLS   #1, CLI$PRESENT
                0B                        50 E9 0007B    BLBC    R0, 3$
                07                   FC   A2 E9 0007E    BLBC    RDB_EXISTS, 3$                         ;  2182
00000000G       00                        00 FB 00082    CALLS   #0, UAF$REMOVE_IDENT_RECBUF            ;  2184
```

```
                                    04 00089 3$:     RET                                    ; 2188

; Routine Size:  138 bytes,    Routine Base:  $CODE$ + 0B61
```

```
;  2112     2189  1 %sbttl 'remove_proxy - remove a proxy record'
;  2113     2190  1 global routine remove_proxy : novalue =
;  2114     2191  2 begin
;  2115     2192  2
;  2116     2193  2 !++
;  2117     2194  2 !
;  2118     2195  2 ! FUNCTIONAL CHARACTERISTICS:
;  2119     2196  2 !
;  2120     2197  2 !       This routine removes proxy login entries from NETUAF.DAT
;  2121     2198  2 !
;  2122     2199  2 ! INPUTS:
;  2123     2200  2 !
;  2124     2201  2 !       none
;  2125     2202  2 !
;  2126     2203  2 ! OUTPUTS:
;  2127     2204  2 !
;  2128     2205  2 !       none
;  2129     2206  2 !
;  2130     2207  2 ! IMPLICIT INPUTS:
;  2131     2208  2 !
;  2132     2209  2 !       none
;  2133     2210  2 !
;  2134     2211  2 ! IMPLICIT OUTPUTS:
;  2135     2212  2 !
;  2136     2213  2 !       none
;  2137     2214  2 !
;  2138     2215  2 ! ROUTINE VALUE:
;  2139     2216  2 !
;  2140     2217  2 !       none
;  2141     2218  2 !
;  2142     2219  2 ! SIDE EFFECTS:
;  2143     2220  2 !
;  2144     2221  2 !       An entry is removed from NETUAF.DAT
;  2145     2222  2 !
;  2146     2223  2 !--
;  2147     2224  2
;  2148     2225  2 local
;  2149     2226  2         node_len,
;  2150     2227  2         node_ptr,
;  2151     2228  2         remuser_len,
;  2152     2229  2         remuser_ptr,
;  2153     2230  2         counter,
;  2154     2231  2         success;
;  2155     2232  2
;  2156     2233  2 !
;  2157     2234  2 ! Make sure NETUAF.DAT exists
;  2158     2235  2 !
;  2159     2236  2 if not .netuaf_exists
;  2160     2237  2 then return LIB$SIGNAL(UAF$_NAFDNE);
;  2161     2238  2
;  2162     2239  2 !
;  2163     2240  2 ! Retrieve remote name in node::remotename form
;  2164     2241  2 !
;  2165     2242  2 cli$get_value (sd_token1, tokendsc);
;  2166     2243  2
;  2167     2244  2 !
;  2168     2245  2 ! Verify proper format
```

```
;  2169          2246  2 !
;  2170          2247  2 if not remote_parse (node_ptr, node_len, remuser_ptr, remuser_len)
;  2171          2248  2 then return;
;  2172          2249  2
;  2173          2250  2 !
;  2174          2251  2 ! Copy into appropriate fields
;  2175          2252  2 !
;  2176          2253  2 ch$copy (.node_len, .node_ptr, ' ', naf$s_node, netbuf[naf$t_node]);
;  2177          2254  2 ch$copy (.remuser_len, .remuser_ptr, ' ', naf$s_remuser, netbuf[naf$t_remuser]);
;  2178          2255  2
;  2179          2256  2 nafrab[rab$l_rop] = rab$m_rlk;
;  2180          2257  2
;  2181          2258  2 success = get_proxy_record ();
;  2182          2259  2
;  2183          2260  2 !
;  2184          2261  2 ! Delete the record
;  2185          2262  2 !
;  2186          2263  2 if .success
;  2187          2264  2 then
;  2188          2265  3     begin
;  2189          2266  4     if rmsbad ($delete (rab = nafrab))
;  2190          2267  4     then
;  2191          2268  5         LIB$SIGNAL(UAF$_REMERR, 0, .rmserr)
;  2192          2269  4     else
;  2193          2270  5         begin
;  2194          2271  5         security_audit (nsa$k_recid_netuaf_del);
;  2195          2272  5         LIB$SIGNAL(UAF$_PREMMSG);
;  2196          2273  5         end;
;  2197          2274  3     end
;  2198          2275  2 else
;  2199          2276  2     LIB$SIGNAL(UAF$_REMERR);
;  2200          2277  2
;  2201          2278  2 netuaf_modified = true;
;  2202          2279  1 end;


                                    01FC 00000          .ENTRY  REMOVE_PROXY, Save R2,R3,R4,R5,R6,R7,R8     ; 2190
                          58 00000000G  8F   D0 00002   MOVL    #UAF$_REMERR, R8                            ;
                          57 00000000G  00   9E 00009   MOVAB   LIB$SIGNAL, R7                              ;
                          56 00000000'  00   9E 00010   MOVAB   RMSERR, R6                                  ;
                          5E            10   C2 00017   SUBL2   #16, SP                                    ;
                          0A      F8    A6   E8 0001A   BLBS    NETUAF_EXISTS, 1$                           ; 2234
                             00000000G  8F   DD 0001E   PUSHL   #UAF$_NAFDNE                                ; 2237
                          67            01   FB 00024   CALLS   #1, LIB$SIGNAL                            ;
                                        04   00027   RET                                                  ;
                                E8      A6   9F 00028 1$: PUSHAB  TOKENDSC                                ; 2242
                             00000000'  00   9F 0002B   PUSHAB  SD_TOKEN1                                  ;
                00000000G   00          02   FB 00031   CALLS   #2, CLI$GET_VALUE                         ;
                                        5E   DD 00038   PUSHL   SP                                       ; 2247
                                   08   AE   9F 0003A   PUSHAB  REMUSER_PTR                               ;
                                   10   AE   9F 0003D   PUSHAB  NODE_LEN                                  ;
                                   18   AE   9F 00040   PUSHAB  NODE_PTR                                  ;
                             FA17  CF         04   FB 00043   CALLS   #4, REMOTE_PARSE                    ;
                          63               50   E9 00048   BLBC    R0, 6$                               ;
```

```
                20              20        0C    BE        08    AE  2C 0004B                MOVC5   NODE_LEN, @NODE_PTR, #32, #32, NETBUF      ; 2253
                                                      058C    C6      00052
                20              20        04    BE              6E  2C 00055                MOVC5   REMUSER_LEN, @REMUSER_PTR, #32, #32, -     ; 2254
                                                      05AC    C6      0005B                           NETBUF+32
                                    0638    C6 00080000       8F  D0 0005E                MOVL    #524288, NAFRAB+4                          ; 2256
                          00000000V 00                        00  FB 00067                CALLS   #0, GET_PROXY_RECORD                       ; 2258
                                    31                        50  E9 0006E                BLBC    SUCCESS, 3$                                ; 2263
                                                      0634    C6  9F 0007'                PUSHAB  NAFRAB                                     ; 2266
                          00000000G 00                        01  FB 00075                CALLS   #1, SYS$DELETE
                                    66                        50  D0 0007C                MOVL    R0, RMSERR
                                    0B                        50  E8 0007F                BLBS    R0, 2$
                                                              66  DD 00082                PUSHL   RMSERR                                     ; 2268
                                                              7E  D4 00084                CLRL    -(SP)
                                                              58  DD 00086                PUSHL   R8
                                    67                        03  FB 00088                CALLS   #3, LIB$SIGNAL
                                                              1A  11 0008B                BRB     5$
                                              00020003        8F  DD 0008D 2$:            PUSHL   #131075                                    ; 2271
                          00000000V 00                        01  FB 00093                CALLS   #1, SECURITY_AUDIT
                                              00000000G       8F  DD 0009A                PUSHL   #UAF$_PREMMSG                              ; 2272
                                                              02  11 000A0                BRB     4$                                         ; 2276
                                                              58  DD 000A2 3$:            PUSHL   R8
                                    67                        01  FB 000A4 4$:            CALLS   #1, LIB$SIGNAL
                          00000000' 00                        01  D0 000A7 5$:            MOVL    #1, NETUAF_MODIFIED                        ; 2278
                                                                  04 000AE 6$:            RET                                                ; 2279

; Routine Size.  175 bytes,    Routine Base:  $CODE$ + 0BEB
```

```
;  2204     2280  1 %sbttl 'rename_uaf - rename user record'
;  2205     2281  1 global routine rename_uaf : novalue =
;  2206     2282  2 begin
;  2207     2283  2
;  2208     2284  2 !++
;  2209     2285  2 !
;  2210     2286  2 ! FUNCTIONAL DESCRIPTION:
;  2211     2287  2 !
;  2212     2288  2 !       Effect a user authorization record rename, by
;  2213     2289  2 !       performing a COPY and a REMOVE operation.
;  2214     2290  2 !
;  2215     2291  2 ! INPUTS:
;  2216     2292  2 !
;  2217     2293  2 !       none
;  2218     2294  2 !
;  2219     2295  2 ! IMPLICIT INPUTS:
;  2220     2296  2 !
;  2221     2297  2 !       none
;  2222     2298  2 !
;  2223     2299  2 ! OUTPUTS:
;  2224     2300  2 !
;  2225     2301  2 !       none
;  2226     2302  2 !
;  2227     2303  2 ! IMPLICIT OUTPUTS:
;  2228     2304  2 !
;  2229     2305  2 !       none
;  2230     2306  2 !
;  2231     2307  2 ! SIDE EFFECTS:
;  2232     2308  2 !
;  2233     2309  2 !       A user authorization record is copied with a newname; the
;  2234     2310  2 !       original record is then deleted.
;  2235     2311  2 !       This routine also causes updating of any NETUAF entries for the
;  2236     2312  2 !       local user which is to be renamed.
;  2237     2313  2 !
;  2238     2314  2 !--
;  2239     2315  2
;  2240     2316  2 uaf$gl_ctlmsk[uaf$v_rename] = true;
;  2241     2317  2 !
;  2242     2318  2 ! Copy the new authorization record
;  2243     2319  2 !
;  2244     2320  3 if  (copy_uaf ())
;  2245     2321  2 then
;  2246     2322  3     begin
;  2247     2323  3 !
;  2248     2324  3 ! Remove the old authorization record
;  2249     2325  3 !
;  2250     2326  3     remove_uaf ();
;  2251     2327  3     LIB$SIGNAL(UAF$_RENMSG);
;  2252     2328  3 !
;  2253     2329  3 ! Because passwords are folded in with the username, passwords for
;  2254     2330  3 ! RENAMEd records will no longer work--warn the user
;  2255     2331  3 !
;  2256     2332  3     if .pwd_flag
;  2257     2333  3     then LIB$SIGNAL(UAF$_DEFPWD);
;  2258     2334  3
;  2259     2335  3 !
;  2260     2336  3 ! Modify the rights data base if one exists
```

```
;  2261          2337  3 !
;  2262          2338  4     if (cli$present ( sd_modify_identifier ) and
;  2263          2339  4         .rdb_exists )
;  2264          2340  3     then
;  2265          2341  4         begin
;  2266          2342  4         local
;  2267          2343  4             status              : long,
;  2268          2344  4             old_name_buff       : vector [kgb$s_name, byte],
;  2269          2345  4             old_name_desc       : statdesc ;
;  2270          2346  4             new_name_desc       : statdesc ;
;  2271          2347  4
;  2272          2348  4         old_name_desc[length]  = kgb$s_name ;
;  2273          2349  4         old_name_desc[pointer] = old_name_buff ;
;  2274          2350  4         new_name_desc[length]  = .newuser[en ;
;  2275          2351  4         new_name_desc[pointer] = newusername ;
;  2276          2352  4
;  2277          2353  4         uaf$find_uic () ;       !  Build Identifier from the recbuf
;  2278          2354  4
;  2279          2355  4         !
;  2280          2356  4         ! Find the ascii name
;  2281          2357  4         !
;  2282        P 2358  4         status = $idtoasc ( id      = .ident,
;  2283        P 2359  4                             namlen = old_name_desc[length],
;  2284          2360  4                             nambuf = old_name_desc ) ;
;  2285          2361  4         if not .status
;  2286          2362  4         then LIB$SIGNAL(UAF$_RDBMDFYERRU, 2,
;  2287          2363  4                         .ident[uic$v_group],
;  2288          2364  4                         .ident[uic$v_member], .status)
;  2289          2365  4         else
;  2290          2366  5             begin
;  2291          2367  5
;  2292        P 2368  5             status = $mod_ident ( id       = .ident,
;  2293          2369  5                                   new_name = new_name_desc ) ;
;  2294          2370  5
;  2295          2371  5             if not .status
;  2296          2372  5                 then LIB$SIGNAL(UAF$_RDBMDFYERR, 1, old_name_desc, .status)
;  2297          2373  5                 else
;  2298          2374  6                     begin
;  2299          2375  6                     rightslist_modified = true ;
;  2300          2376  6                     LIB$SIGNAL(UAF$_RDBMDFYMSG, 1, old_name_desc);
;  2301          2377  5                     end ;
;  2302          2378  4             end ;
;  2303          2379  3         end ;
;  2304          2380  3
;  2305          2381  2     end;
;  2306          2382  2 !
;  2307          2383  2 ! Reset flags
;  2308          2384  2 !
;  2309          2385  2 rename_ph2 = false;
;  2310          2386  2 uaf$gl_ctlmsk[uaf$v_rename] = false;
;  2311          2387  2
;  2312          2388  1 end;
```

```
                                 007C 00000          .ENTRY  RENAME_UAF, Save R2,R3,R4,R5,R6                 ; 2281
                          56 00000000'   00  9E 00002        MOVAB   NEWUSERLEN, R6
                          55 00000000G   00  9E 00009        MOVAB   IDENT, R5
                          54 00000000'   00  9E 00010        MOVAB   UAF$GL_CTLMSK, R4
                          53 00000000G   00  9E 00017        MOVAB   LIB$SIGNAL, R3
                          5E             30  C2 0001E        SUBL2   #48, SP
                          64             01  88 00021        BISB2   #1, UAF$GL_CTLMSK                       ; 2316
                    FA08  CF             00  FB 00024        CALLS   #0, COPY_UAF                            ; 2320
                          29             50  E9 00029        BLBC    R0, 2$
                    FE96  CF             00  FB 0002C        CALLS   #0, REMOVE_UAF                          ; 2326
                          00000000G      8F  DD 00031        PUSHL   #UAF$_RENMSG                            ; 2327
                          63             01  FB 00037        CALLS   #1, LIB$SIGNAL
                          09       06F4  C4  E9 0003A        BLBC    PWD_FLAG, 1$                            ; 2332
                          00000000G      8F  DD 0003F        PUSHL   #UAF$_DEFPWD                            ; 2333
                          63             01  FB 00045        CALLS   #1, LIB$SIGNAL
                          00000000'      00  9F 00048 1$:    PUSHAB  SD_MODIFY_IDENTIFIER                    ; 2338
                00000000G 00             01  FB 0004E        CALLS   #1, CLI$PRESENT
                          60             50  E9 00055 2$:    BLBC    R0, 3$
                          5C         70  A4  E9 00058        BLBC    RDB_EXISTS, 3$                          ; 2339
                       08 AE 010E0000    8F  D0 0005C        MOVL    #17894720, OLD_NAME_DESC                ; 2345
                                    0C   AE  D4 00064        CLRL    OLD_NAME_DESC+4
                          6E 010E0000    8F  D0 00067        MOVL    #17894720, NEW_NAME_DESC                ; 2346
                                    04   AE  D4 0006E        CLRL    NEW_NAME_DESC+4
                       08 AE            20  B0 00071        MOVW    #32, OLD_NAME_DESC                      ; 2348
                       0C AE        10  AE  9E 00075        MOVAB   OLD_NAME_BUFF, OLD_NAME_DESC+4          ; 2349
                          6E            66  B0 0007A        MOVW    NEWUSERLEN, NEW_NAME_DESC               ; 2350
                       04 AE        04  A6  9E 0007D        MOVAB   NEWUSERNAME, NEW_NAME_DESC+4            ; 2351
                00000000G 00             00  FB 00082        CALLS   #0, UAF$FIND_UIC                        ; 2353
                                         7E  7C 00089        CLRG    -(SP)                                   ; 2360
                                         7E  D4 0008B        CLRL    -(SP)
                                     14  AE  9F 0008D        PUSHAB  OLD_NAME_DESC
                                     18  AE  9F 00090        PUSHAB  OLD_NAME_DESC
                                         65  DD 00093        PUSHL   IDENT
                          00000000G      00  06 FB 00095        CALLS   #6, SYS$IDTOASC
                                         52  50 DD 0009C        MOVL    R0, STATUS
                                         18  52 E8 0009F        BLBS    STATUS, 4$                              ; 2361
                                         52  DD 000A2        PUSHL   STATUS                                  ; 2364
                                         7E  65 3C 000A4        MOVZWL  IDENT, -(SP)
                    7E          02   A5      0E  00 EF 000A7        EXTZV   #0, #14, IDENT+2, -(SP)             ; 2363
                                         02  DD 000AD        PUSHL   #2                                      ; 2362
                                 00000000G   8F  DD 000AF        PUSHL   #UAF$_RDBMDFYERRU
                                         63  05 FB 000B5        CALLS   #5, LIB$SIGNAL
                                             3A 11 000B8 3$:     BRB     6$
                                             7E D4 000BA 4$:     CLRL    -(SP)                               ; 2369
                                     04  AE  9F 000BC        PUSHAB  NEW_NAME_DESC
                                         7E  7C 000BF        CLRQ    -(SP)
                                         65  DD 000C1        PUSHL   IDENT
                          00000000G      00  05 FB 000C3        CALLS   #5, SYS$MOD_IDENT
                                         52  50 DD 000CA        MOVL    R0, STATUS
                                         12  52 E8 000CD        BLBS    STATUS, 5$                              ; 2371
                                         52  DD 000D0        PUSHL   STATUS                                  ; 2372
                                     0C  AE  9F 000D2        PUSHAB  OLD_NAME_DESC
                                         01  DD 000D5        PUSHL   #1
                                 00000000G   8F  DD 000D7        PUSHL   #UAF$_RDBMDFYERR
                                         63  04 FB 000DD        CALLS   #4, LIB$SIGNAL
                                             12 11 000E0        BRB     6$
                                 78  A4      01  90 000E2 5$:    MOVB    #1, RIGHTSLIST_MODIFIED             ; 2375
```

```
                                 08  AE  9F 000E6          PUSHAB  OLD_NAME_DESC                    ; 2376
                                 01  DD 000E9              PUSHL   #1                              ;
                      00000000G  8F  DD 000EB              PUSHL   #UAF$_RDBMDFYMSG                ;
                             63  03  FB 000F1              CALLS   #3, LIB$SIGNAL                  ;
                                 D8  A6  94 000F4 6$:      CLRB    RENAME_PH2                      ; 2385
                             64  01  8A 000F7              BICB2   #1, UAF$GL_CTLMSK               ; 2386
                                     04 000FA              RET                                     ; 2388

; Routine Size:  251 bytes,    Routine Base:  $CODE$ + 0C9A
```

```
;  2314     2389  1 %sbttl 'adjust_proxy - implicitly remove/update proxy record'
;  2315     2390  1 global routine adjust_proxy (remove) : novalue =
;  2316     2391  2 begin
;  2317     2392  2
;  2318     2393  2 !++
;  2319     2394  2 !
;  2320     2395  2 ! FUNCTIONAL DESCRIPTION:
;  2321     2396  2 !
;  2322     2397  2 !       This routine performs the operations implicitly indicated by
;  2323     2398  2 !       REMOVE or RENAME operations on SYSUAF.DAT.  If a SYSUAF.DAT
;  2324     2399  2 !       record is removed, then any corresponding NETUAF.DAT entries must
;  2325     2400  2 !       also be deleted.  If a SYSUAF.DAT record is renamed, then any
;  2326     2401  2 !       corresponding NETUAF.DAT entries must be updated.
;  2327     2402  2 !
;  2328     2403  2 ! INPUTS:
;  2329     2404  2 !
;  2330     2405  2 !       remove - a flag which indicates that the NETUAF.DAT record
;  2331     2406  2 !                should be removed.  If false, then the record
;  2332     2407  2 !                should be updated.
;  2333     2408  2 !
;  2334     2409  2 ! OUTPUTS:
;  2335     2410  2 !
;  2336     2411  2 !       none
;  2337     2412  2 !
;  2338     2413  2 ! IMPLICIT INPUTS:
;  2339     2414  2 !
;  2340     2415  2 !       olduserlen  - the old username length for a RENAME
;  2341     2416  2 !       oldusername - the old username string for a RENAME
;  2342     2417  2 !       TOKENPTR - the new username for RENAME, the removed username for REMOVE
;  2343     2418  2 !       TOKENLEN - "    "    "      length "  ",    "      "        length " "
;  2344     2419  2 !
;  2345     2420  2 ! IMPLICIT OUTPUTS:
;  2346     2421  2 !
;  2347     2422  2 !       none
;  2348     2423  2 !
;  2349     2424  2 ! SIDE EFFECTS:
;  2350     2425  2 !
;  2351     2426  2 !        A  record is removed/updated in NETUAF.DAT
;  2352     2427  2 !
;  2353     2428  2 !--
;  2354     2429  2
;  2355     2430  2 local
;  2356     2431  2         modified: initial(false),
;  2357     2432  2         status;
;  2358     2433  2
;  2359     2434  2 !
;  2360     2435  2 ! Set access to sequential because we need to check for
;  2361     2436  2 ! multiple entries
;  2362     2437  2 !
;  2363     2438  2 nafrab[rab$l_rop] = rab$m_rlk;          ! Lock records for writing
;  2364     2439  2 nafrab[rab$b_rac] = rab$c_seq;
;  2365     2440  2 $rewind (rab = nafrab);
;  2366     2441  2
;  2367     2442  2 !
;  2368     2443  2 ! Until EOF...
;  2369     2444  2 !
;  2370     2445  2 while status = get_proxy_record ()
```

```
;  2371          2446  2 do
;  2372          2447        begin
;  2373          2448        local
;  2374          2449            locuser_len,
;  2375          2450            blank_ptr;
;  2376          2451
;  2377          2452    !
;  2378          2453    ! Find end of user name...
;  2379          2454    !
;  2380          2455    !
;  2381          2456    ! If not found in 12 characters, it must be the full 12
;  2382          2457    ! characters in length
;  2383          2458    !
;  2384          2459        if ch$fail (blank_ptr = ch$find_ch (naf$s_localuser,
;  2385          2460                                            netbuf[naf$t_localuser], ' '))
;  2386          2461        then
;  2387          2462            locuser_len = naf$s_localuser
;  2388          2463        else
;  2389          2464            locuser_len = .blank_ptr - netbuf[naf$t_localuser];
;  2390          2465
;  2391          2466    !
;  2392          2467    ! If this is a record to be removed, delete it
;  2393          2468    !
;  2394          2469        if .remove
;  2395          2470        then
;  2396          2471  4         begin
;  2397          2472  4         if .tokenlen eql .locuser_len
;  2398          2473  4         and  ch$eql (.tokenlen, .tokenptr, .locuser_len, netbuf[naf$t_localuser])
;  2399          2474  4         then
;  2400          2475  5             begin
;  2401          2476  5             netuaf_modified = true;
;  2402          2477  5             $delete (rab = nafrab);
;  2403          2478  5             security_audit (nsa$k_recid_netuaf_del);
;  2404          2479  4             end;
;  2405          2480  4         end
;  2406          2481  4    !
;  2407          2482  4    ! otherwise, change the localusername field to reflect the
;  2408          2483  4    ! new username in SYSUAF.DAT
;  2409          2484  4    !
;  2410          2485  3        else
;  2411          2486  4         begin
;  2412          2487  4         modified = true;
;  2413          2488  4         if .olduserlen eql .locuser_len
;  2414          2489  4         and  ch$eql (.olduserlen, oldusername,
;  2415          2490  4                      .locuser_len, netbuf[naf$t_localuser])
;  2416          2491  4         then
;  2417          2492  5             begin
;  2418          2493  5             ch$copy (uaf$s_username, recbuf[uaf$t_username],' ',
;  2419          2494  5                      naf$s_localuser,netbuf[naf$t_localuser]);
;  2420          2495  5             $update (rab = nafrab);
;  2421          2496  5             netuaf_modified = true;
;  2422          2497  5             security_audit (nsa$k_recid_netuaf_mod, oldusername);
;  2423          2498  4             end;
;  2424          2499  3         end;
;  2425          2500  2     end;
;  2426          2501  2
;  2427          2502  2 if .modified then
```

```
;  2428     2503  2     LIB$SIGNAL(UAF$_ZZPRACREN, 2, .olduserlen, oldusername);
;  2429     2504  2
;  2430     2505  2 !
;  2431     2506  2 ! Return to keyed access
;  2432     2507  2 !
;  2433     2508  2 nafrab[rab$b_rac] = rab$c_key;
;  2434     2509  1 end;


                                                                .EXTRN  SYS$REWIND

                                                07FC 00000      .ENTRY  ADJUST_PROXY, Save R2,R3,R4,R5,R6,R7,R8,R9,-; 2390
                                                                        R10
                                5A 00000000V    00   9E 00002  MOVAB   SECURITY_AUDIT, R10                          ;
                                59 00000000'    00   9E 00009  MOVAB   OLDUSERNAME, R9                            ;
                                58 00000000'    00   9E 00010  MOVAB   NETBUF+64, R8                             ;
                                                56   D4 00017  CLRL    MODIFIED                                  ; 2391
                             6C AB 00080000     8F   D0 00019  MOVL    #524288, NAFRAB+4                         ; 2438
                                      0086      C8   94 00021  CLRB    NAFRAB+30                                 ; 2439
                                        68      A8   9F 00025  PUSHAB  NAFRAB                                    ; 2440
                   00000000G   00               01   FB 00028  CALLS   #1, SYS$REWIND                           ;
                   00000000V   00               00   FB 0002F  1$:    CALLS   #0, GET_PROXY_RECORD              ; 2445
                               57               50   D0 00036  MOVL    R0, STATUS                                ;
                               03               57   E8 00039  BLBS    STATUS, 2$                                ;
                                              0081   31 0003C  BRW     8$                                        ;
                         68                    20   3A 0003F  2$:    LOCC    #32, #32, NETBUF+64               ; 2460
                                               02   12 00043  BNEQ    3$                                        ;
                                               51   D4 00045  CLRL    R1                                        ;
                                               51   D5 00047  3$:    TSTL    BLANK_PTR                         ;
                                               05   12 00049  BNEQ    4$                                        ;
                                     50        20   D0 0004B  MOVL    #32, LOCUSER_LEN                          ; 2462
                                               07   11 0004E  BRB     5$                                        ;
                                     52        68   9E 00050  4$:    MOVAB   NETBUF+64, R2                     ; 2464
                            50       51        52   C3 00053  SUBL3   R2, BLANK_PTR, LOCUSER_LEN                ;
                                     30  04    AC   E9 00057  5$:    BLBC    REMOVE, 7$                        ; 2472
                                     51  FA1C  C8   3C 0005B  MOVZWL  TOKENLEN, R1                              ;
                                     50        51   D1 00060  CMPL    R1, LOCUSER_LEN                           ;
                                               CA   12 00063  BNEQ    1$                                        ;
                                     52  FA20  C8   D0 00065  MOVL    TOKENPTR, R2                              ; 2473
                   50        00      62        51   2D 0006A  CMPC5   R1, (R2), #0, LOCUSER_LEN, NETBUF+64     ;
                                                68   0006F                                                      ;
                                               BD   12 00070  BNEQ    1$                                        ;
                                  F4   A9      01   D0 00072  MOVL    #1, NETUAF_MODIFIED                       ; 2476
                                        68     A8   9F 00076  PUSHAB  NAFRAB                                    ; 2477
                            00000000G   00     01   FB 00079  CALLS   #1, SYS$DELETE                           ;
                                     00020003  8F   DD 00080  PUSHL   #131075                                   ; 2478
                                     6A        01   FB 00086  CALLS   #1, SECURITY_AUDIT                        ;
                                               A4   11 00089  6$:    BRB     1$                                ; 2469
                                     56        01   D0 0008B  7$:    MOVL    #1, MODIFIED                      ; 2487
                                     51  FC    A9   D0 0008E  MOVL    OLDUSERLEN, R1                            ; 2488
                                     50        51   D1 00092  CMPL    R1, LOCUSER_LEN                           ;
                                               98   12 00095  BNEQ    1$                                        ;
                   50        00      69        51   2D 00097  CMPC5   R1, OLDUSERNAME, #0, LOCUSER_LEN, NETBUF+64 ; 2490
                                                68   0009C                                                      ;
                                               90   12 0009D  BNEQ    1$                                        ;
                            68 FA40  C8        20   28 0009F  MOVC3   #32, RECBUF+4, NETBUF+64                  ; 2494
```

```
                                 68   A8  9F 000A5          PUSHAB  NAFRAB                                  ; 2495
                  00000000G      00   01  FB 000A8          CALLS   #1, SYS$UPDATE                         ;
                         F4      A9   01  D0 000AF          MOVL    #1, NETUAF_MODIFIED                    ; 2496
                                      59  DD 000B3          PUSHL   R9                                     ; 2497
                           00030003   8F  DD 000B5          PUSHL   #196611                                ;
                                 6A   02  FB 000BB          CALLS   #2, SECURITY_AUDIT                     ;
                                      C9  11 000BE          BRB     6$                                     ; 2445
                                 14   56  E9 000C0 8$:      BLBC    MODIFIED, 9$                           ; 2502
                                      59  DD 000C3          PUSHL   R9                                     ; 2503
                                 FC   A9  DD 000C5          PUSHL   OLDUSERLEN                             ;
                                      02  DD 000C8          PUSHL   #2                                     ;
                           00000000G  8F  DD 000CA          PUSHL   #UAF$_ZZPRACREN                        ;
                  00000000G      00   04  FB 000D0          CALLS   #4, LIB$SIGNAL                         ;
                       0086      C8   01  90 000D7 9$:      MOVB    #1, NAFRAB+30                          ; 2508
                                      04  000DC            RET                                            ; 2509

; Routine Size:  221 bytes,    Routine Base:  $CODE$ + 0D95
```

```
;  2436     2510  1 %sbttl 'default_uaf - change default record'
;  2437     2511  1 global routine default_uaf : novalue =
;  2438     2512  2 begin
;  2439     2513  2
;  2440     2514  2 !++
;  2441     2515  2 !
;  2442     2516  2 ! FUNCTIONAL DESCRIPTION:
;  2443     2517  2 !
;  2444     2518  2 !       Change a default value in the default record.
;  2445     2519  2 !
;  2446     2520  2 ! INPUTS:
;  2447     2521  2 !
;  2448     2522  2 !       none
;  2449     2523  2 !
;  2450     2524  2 ! IMPLICIT INPUTS:
;  2451     2525  2 !
;  2452     2526  2 !       none
;  2453     2527  2 !
;  2454     2528  2 ! OUTPUTS:
;  2455     2529  2 !
;  2456     2530  2 !       none
;  2457     2531  2 !
;  2458     2532  2 ! IMPLICIT OUTPUTS:
;  2459     2533  2 !
;  2460     2534  2 !       none
;  2461     2535  2 !
;  2462     2536  2 ! ROUTINE VALUE:
;  2463     2537  2 !
;  2464     2538  2 !       none
;  2465     2539  2 !
;  2466     2540  2 ! SIDE EFFECTS:
;  2467     2541  2 !
;  2468     2542  2 !       none
;  2469     2543  2 !--
;  2470     2544  2
;  2471     2545  2
;  2472     2546  2 !
;  2473     2547  2 ! Locate default record and load it into RECBUF
;  2474     2548  2 ! if not already there (via an indirect MODIFY DEFAULT)
;  2475     2549  2 !
;  2476     2550  2
;  2477     2551  2 if not .mod_default
;  2478     2552  2 then
;  2479     2553  3     begin
;  2480     2554  3     if not locate_user (.defuser<0,8>, defuser+1, true)
;  2481     2555  3     then
;  2482     2556  4         begin
;  2483     2557  4         LIB$SIGNAL(UAF$_DEFERR, 0, .rmserr);
;  2484     2558  4         return;
;  2485     2559  3         end;
;  2486     2560  2     end;
;  2487     2561  2 !
;  2488     2562  2 ! The encrypted password field of the DEFAULT record can not be propagated
;  2489     2563  2 ! to another user, because the encryption algorithm takes the user name as
;  2490     2564  2 ! an input.  The user is merely warned that this qualifier has no effect.
;  2491     2565  2 !
;  2492     2566  2
```

```
;  2493          2567  2 pwd_flag = true;
;  2494          2568  2
;  2495          2569  2 !
;  2496          2570  2 ! Update values supplied and exit if errors.
;  2497          2571  2 !
;  2498          2572  2
;  2499          2573  2 if update_record ()
;  2500          2574  2 then
;  2501          2575  3     begin
;  2502          2576  3     if not .pwd_flag
;  2503          2577  3     then LIB$SIGNAL(UAF$_NODEFPWD);
;  2504          2578  3
;  2505          2579  3 !
;  2506          2580  3 ! Now write the modified record back into the DEFAULT_RECORD buffer.
;  2507          2581  3 !
;  2508          2582  3     default_size = .uafrab[rab$w_rsz];
;  2509          2583  3     ch$move (.default_size, recbuf, default_record);
;  2510          2584  3
;  2511          2585  3 !
;  2512          2586  3 ! Update the default record in the file.  Note that file has changed.
;  2513          2587  3 !
;  2514          2588  4     if not rmsok ($update (rab = uafrab))
;  2515          2589  3     then
;  2516          2590  3         LIB$SIGNAL(UAF$_MDFYERR, 0, .rmserr)
;  2517          2591  3     else
;  2518          2592  4         begin
;  2519          2593  4         modify_flag = true;
;  2520          2594  4         security_audit (nsa$k_recid_sysuaf_mod);
;  2521          2595  4         if not .mod_default
;  2522          2596  4         then
;  2523          2597  4             LIB$SIGNAL(UAF$_MDFYMSG)
;  2524          2598  4         else
;  2525          2599  4             mod_default = false;
;  2526          2600  3         end;
;  2527          2601  3     end
;  2528          2602  3
;  2529          2603  2 else
;  2530          2604  2     $release (rab = uafrab);                    ! unlock the record
;  2531          2605  2
;  2532          2606  1 end;


                                    01FC 00000          .ENTRY  DEFAULT_UAF, Save R2,R3,R4,R5,R6,R7,R8      ; 2511
                5E 00000000G    00  9E 00002          MOVAB   LIB$SIGNAL, R8                              ;
                57 00000000'    00  9E 00009          MOVAB   MOD_DEFAULT, R7                             ;
                56 00000000'    00  9E 00010          MOVAB   RMSERR, R6                                  ;
                25              67  E8 00017          BLBS    MOD_DEFAULT, 1$                             ; 2551
                                01  DD 0001A          PUSHL   #1                                          ; 2554
                   00000000'    00  9F 0001C          PUSHAB  DEFUSER+1                                   ;
                7E 00000000'    00  9A 00022          MOVZBL  DEFUSER, -(SP)                              ;
          00000000V 00          03  FB 00029          CALLS   #3, LOCATE_USER                             ;
                0C              50  E8 00030          BLBS    R0, 1$                                      ;
                                66  DD 00033          PUSHL   RMSERR                                      ; 2557
                                7E  D4 00035          CLRL    -(SP)                                       ;
```

```
                                 00000000G  8F  DD 00037          PUSHL   #UAF$_DEFERR                                    ;
                                             48  11 0003D          BRB     3$                                              ;
                           0680  C6                  01  D0 0003F 1$:    MOVL    #1, PWD_FLAG                            ; 2567
                  00000000G  00                      00  FB 00044        CALLS   #0, UPDATE_RECORD                       ; 2573
                             5E                      50  E9 0004B        BLBC    R0, 6$                                  ;
                             09       0680           C6  E8 0004E        BLBS    PWD_FLAG, 2$                            ; 2576
                                 00000000G           8F  DD 00053        PUSHL   #UAF$_NODEFPWD                          ; 2577
                             68                      01  FB 00059        CALLS   #1, LIB$SIGNAL                          ;
                     0458    C7       0612           C6  B0 0005C 2$:    MOVW    UAFRAB+34, DEFAULT_SIZE                 ; 2582
           045C  C7    08    A6       0458           C7  28 00063        MOVC3   DEFAULT_SIZE, RECBUF, DEFAULT_RECORD    ; 2583
                             05F0                    C6  9F 0006C        PUSHAB  UAFRAB                                  ; 2588
                  00000000G  00                      01  FB 00070        CALLS   #1, SYS$UPDATE                          ;
                             66                      50  D0 00077        MOVL    R0, RMSERR                              ;
                             0E                      50  E8 0007A        BLBS    R0, 4$                                  ;
                                                     66  DD 0007D        PUSHL   RMSERR                                  ; 2590
                                                     7E  D4 0007F        CLRL    -(SP)                                   ;
                                 00000000G           8F  DD 00081        PUSHL   #UAF$_MDFYERR                           ;
                             68                      03  FB 00087 3$:    CALLS   #3, LIB$SIGNAL                          ;
                                                         04 0008A        RET                                             ;
                       04    A7                      01  D0 0008B 4$:    MOVL    #1, MODIFY_FLAG                         ; 2593
                                 00030002            8F  DD 0008F        PUSHL   #196610                                 ; 2594
                  00000000V  00                      01  FB 00095        CALLS   #1, SECURITY_AUDIT                      ;
                             0A                      67  E8 0009C        BLBS    MOD_DEFAULT, 5$                         ; 2595
                                 00000000G           8F  DD 0009F        PUSHL   #UAF$_MDFYMSG                           ; 2597
                             68                      01  FB 000A5        CALLS   #1, LIB$SIGNAL                          ;
                                                         04 000A8        RET                                             ;
                                                     67  D4 000A9 5$:    CLRL    MOD_DEFAULT                             ; 2599
                                                         04 000AB        RET                                             ; 2573
                             05F0                    C6  9F 000AC 6$:    PUSHAB  UAFRAB                                  ; 2604
                  00000000G  00                      01  FB 000B0        CALLS   #1, SYS$RELEASE                         ;
                                                         04 000B7        RET                                             ; 2606

; Routine Size:  184 bytes,    Routine Base:  $CODE$ + 0E72
```

```
;  2534     2607  1 %sbttl 'list_proxy - list proxy entries in NETUAF.LIS'
;  2535     2608  1 global routine list_proxy : novalue =
;  2536     2609  2 begin
;  2537     2610  2
;  2538     2611  2 !++
;  2539     2612  2 !
;  2540     2613  2 ! FUNCTIONAL DESCRIPTION:
;  2541     2614  2 !
;  2542     2615  2 !       This routine produces a listing of the entire NETUAF.DAT file
;  2543     2616  2 !       in NETUAF.LIS.
;  2544     2617  2 !
;  2545     2618  2 ! INPUTS:
;  2546     2619  2 !
;  2547     2620  2 !       none
;  2548     2621  2 !
;  2549     2622  2 ! OUTPUTS:
;  2550     2623  2 !
;  2551     2624  2 !       none
;  2552     2625  2 !
;  2553     2626  2 ! SIDE EFFECTS:
;  2554     2627  2 !
;  2555     2628  2 !       A listing file is produced
;  2556     2629  2 !
;  2557     2630  2 !--
;  2558     2631  2
;  2559     2632  2 local
;  2560     2633  2         action;
;  2561     2634  2
;  2562     2635  2 !
;  2563     2636  2 ! Make sure NETUAF.DAT exists
;  2564     2637  2 !
;  2565     2638  2 if not .netuaf_exists
;  2566     2639  2 then return LIB$SIGNAL(UAF$_NAFDNE);
;  2567     2640  2
;  2568     2641  2 !
;  2569     2642  2 ! Set up the listing file FAB and connect RAB
;  2570     2643  2 !
;  2571     2644  2 nlstfab[fab$v_dlt] = false;
;  2572     2645  3 if rmsbad ($create (fab = nlstfab))
;  2573     2646  2 then return LIB$SIGNAL(UAF$_LSTERR, 0, .rmserr);
;  2574     2647  2
;  2575     2648  3 if rmsbad ($connect (rab = nlstrab))
;  2576     2649  2 then return LIB$SIGNAL(UAF$_LSTERR, 0, .rmserr);
;  2577     2650  2
;  2578     2651  2 !
;  2579     2652  2 ! Set action routine and rab pointer
;  2580     2653  2 !
;  2581     2654  2 wild_netuser = true;
;  2582     2655  2 match_tokenlen = 1;
;  2583     2656  2 match_token = %c'*';
;  2584     2657  2 rabptr = nlstrab;
;  2585     2658  2 action = display_proxy;
;  2586     2659  2 header_flag = true;
;  2587     2660  2 nafrab[rab$l_rop] = rab$m_rrl or rab$m_nlk;
;  2588     2661  2
;  2589     2662  2 LIB$SIGNAL(UAF$_LSTMSG1);
;  2590     2663  2
```

```
;  2591          2664  2 !
;  2592          2665  2 ! LOCATE_PROXY will call the action routine for each proxy record
;  2593          2666  2 !
;  2594          2667  3 if rmsbad (locate_proxy (.action))
;  2595          2668  2 then
;  2596          2669  3     begin
;  2597          2670  3     if .rmserr eql rms$_rnf
;  2598          2671  3     then
;  2599          2672  3         LIB$SIGNAL(UAF$_BADSPC)
;  2600          2673  3     else
;  2601          2674  3         LIB$SIGNAL(UAF$_LSTERR, 0, .rmserr);
;  2602          2675  3     end
;  2603          2676  2 else
;  2604          2677  2     LIB$SIGNAL(UAF$_NETLSTMSG);
;  2605          2678  2
;  2606          2679  2 !
;  2607          2680  2 ! Disconnect RAB and close FAB
;  2608          2681  2 !
;  2609          2682  2 $disconnect (rab = nlstrab);
;  2610          2683  2 $close (fab = nlstfab);
;  2611          2684  2
;  2612          2685  1 end;


                                                                .EXTRN  SYS$DISCONNECT, SYS$CLOSE

                                                007C 00000      .ENTRY  LIST_PROXY, Save R2,R3,R4,R5,R6                 ; 2608
                    56 00000000G   8F   D0 00002        MOVL    #UAF$_LSTERR, R6                                    ;
                    55 00000000G   00   9E 00009        MOVAB   LIB$SIGNAL, R5                                      ;
                    54 00000000'   00   9E 00010        MOVAB   NLSTRAB, R4                                         ;
                    53 00000000'   00   9E 00017        MOVAB   RMSERR, R3                                          ;
                    0A       F8    A3   E8 0001E        BLBS    NETUAF_EXISTS, 1$                                   ; 2638
                          00000000G   8F   DD 00022        PUSHL   #UAF$_NAFDNE                                        ; 2639
                    65                01   FB 00028        CALLS   #1, LIB$SIGNAL                                      ;
                                      04   0002B        RET                                                         ;
                 B5 A4          80   8F   8A 0002C 1$:     BICB2   #128, NLSTFAB+5                                     ; 2644
                               B0   A4   9F 00031        PUSHAB  NLSTFAB                                             ; 2645
          00000000G 00                01   FB 00034        CALLS   #1, SYS$CREATE                                      ;
                    63                50   D0 0003B        MOVL    R0, RMSERR                                          ;
                    0F                50   E9 0003E        BLBC    R0, 2$                                              ;
                                      54   DD 00041        PUSHL   R4                                                  ; 2648
          00000000G 00                01   FB 00043        CALLS   #1, SYS$CONNECT                                     ;
                    63                50   D0 0004A        MOVL    R0, RMSERR                                          ;
                    0A                50   E8 0004D        BLBS    R0, 3$                                              ;
                                      63   DD 00050 2$:     PUSHL   RMSERR                                              ; 2649
                                      7E   D4 00052        CLRL    -(SP)                                               ;
                                      56   DD 00054        PUSHL   R6                                                  ;
                    65                03   FB 00056        CALLS   #3, LIB$SIGNAL                                      ;
                                      04   00059        RET                                                         ;
                 E0 A3                01   D0 0005A 3$:     MOVL    #1, WILD_NETUSER                                    ; 2654
                 DC A3                01   D0 0005E        MOVL    #1, MATCH_TOKENLEN                                  ; 2655
                 98 A3                2A   D0 00062        MOVL    #42, MATCH_TOKEN                                    ; 2656
                 80 A3                64   9E 00066        MOVAB   NLSTRAB, RABPTR                                     ; 2657
                    52 00000000V      00   9E 0006A        MOVAB   DISPLAY_PROXY, ACTION                               ; 2658
               FCC4 C4                01   D0 00071        MOVL    #1, HEADER_FLAG                                     ; 2659
               0638 C3 00100008       8F   D0 00076        MOVL    #1048584, NAFRAB+4                                  ; 2660
```

```
                          00000000G  8F   DD 0007F          PUSHL   #UAF$_LSTMSG1                    ; 2662
                                65   01   FB 00085          CALLS   #1, LIB$SIGNAL                  ;
                                     52   DD 00088          PUSHL   ACTION                          ; 2667
                00000000V       00   01   FB 0008A          CALLS   #1, LOCATE_PROXY                ;
                                63   50   D0 00091          MOVL    R0, RMSERR                      ;
                                1F   50   E8 00094          BLBS    R0, 5$                          ;
                                50   63   D0 00097          MOVL    RMSERR, R0                      ; 2670
                000182B2        8F   50   D1 0009A          CMPL    R0, #98994                      ;
                                     08   12 000A1          BNEQ    4$                              ;
                          00000000G  8F   DD 000A3          PUSHL   #UAF$_BADSPC                    ; 2672
                                     11   11 000A9          BRB     6$                              ;
                                     50   DD 000AB  4$:      PUSHL   R0                              ; 2674
                                     7E   D4 000AD          CLRL    -(SP)                           ;
                                     56   DD 000AF          PUSHL   R6                              ;
                                65   03   FB 000B1          CALLS   #3, LIB$SIGNAL                  ;
                                     09   11 000B4          BRB     7$                              ; 2667
                          00000000G  8F   DD 000B6  5$:      PUSHL   #UAF$_NETLSTMSG                 ; 2677
                                65   01   FB 000BC  6$:      CALLS   #1, LIB$SIGNAL                  ;
                                     54   DD 000BF  7$:      PUSHL   R4                              ; 2682
                00000000G       00   01   FB 000C1          CALLS   #1, SYS$DISCONNECT              ;
                                B0   A4   9F 000C8          PUSHAB  NLSTFAB                         ; 2683
                00000000G       00   01   FB 000CB          CALLS   #1, SYS$CLOSE                   ;
                                     04   00 000D2          RET                                     ; 2685

; Routine Size:  211 bytes,    Routine Base:  $CODE$ + 0F2A
```

```
;  2614     2686  1 %sbttl 'list_uaf - list file routine'
;  2615     2687  1 global routine list_uaf : novalue =
;  2616     2688  2 begin
;  2617     2689  2
;  2618     2690  2 !++
;  2619     2691  2 !
;  2620     2692  2 ! FUNCTIONAL DESCRIPTION:
;  2621     2693  2 !
;  2622     2694  2 !       Display the specified users in a file named 'SYSUAF.LIS'.
;  2623     2695  2 !
;  2624     2696  2 ! INPUTS:
;  2625     2697  2 !
;  2626     2698  2 !       none
;  2627     2699  2 !
;  2628     2700  2 ! IMPLICIT INPUTS:
;  2629     2701  2 !
;  2630     2702  2 !       none
;  2631     2703  2 !
;  2632     2704  2 ! OUTPUTS:
;  2633     2705  2 !
;  2634     2706  2 !       none
;  2635     2707  2 !
;  2636     2708  2 ! IMPLICIT OUTPUTS:
;  2637     2709  2 !
;  2638     2710  2 !       none
;  2639     2711  2 !
;  2640     2712  2 ! ROUTINE VALUE:
;  2641     2713  2 !
;  2642     2714  2 !       none
;  2643     2715  2 !
;  2644     2716  2 ! SIDE EFFECTS:
;  2645     2717  2 !
;  2646     2718  2 !       none
;  2647     2719  2 !--
;  2648     2720  2
;  2649     2721  2 local
;  2650     2722  2         action;
;  2651     2723  2
;  2652     2724  2 !
;  2653     2725  2 ! Obtain the user specification.  This sets wildcard flags and initializes
;  2654     2726  2 ! the appropriate key in RECBUF.
;  2655     2727  2 !
;  2656     2728  2
;  2657     2729  2 if not parse_wild (sd_token1,true)              ! Null string defaults to *
;  2658     2730  2 then return;
;  2659     2731  2
;  2660     2732  2 !
;  2661     2733  2 ! Obtain qualifiers.  This determines which display should be used.
;  2662     2734  2 !
;  2663     2735  2 full_flag = false;
;  2664     2736  2 brief_flag = true;
;  2665     2737  2
;  2666     2738  3 if cli$present (sd_full) or (not cli$present (sd_brief))
;  2667     2739  2 then
;  2668     2740  3     begin
;  2669     2741  3     brief_flag = false;
;  2670     2742  3     full_flag  = true;
```

```
;  2671     2743  2     end;
;  2672     2744  2
;  2673     2745  2 !
;  2674     2746  2 ! Create the listing file.
;  2675     2747  2 !
;  2676     2748  2
;  2677     2749  2 lstfab[fab$v_dlt] = false;                ! initialize DLT bit
;  2678     2750  3 if rmsbad ($create (fab = lstfab))
;  2679     2751  2 then return LIB$SIGNAL(UAF$_LSTERR, 0, .rmserr);
;  2680     2752  2
;  2681     2753  3 if rmsbad ($connect (rab = lstrab))
;  2682     2754  2 then return LIB$SIGNAL(UAF$_LSTERR, 0, .rmserr);
;  2683     2755  2
;  2684     2756  2 !
;  2685     2757  2 ! Request a header record for the file and aim RABPTR at our RAB.
;  2686     2758  2 !
;  2687     2759  2
;  2688     2760  2 header_flag = true;
;  2689     2761  2 rabptr = lstrab;
;  2690     2762  2 found_match = false;
;  2691     2763  2 uafrab[rab$l_rop] = rab$m_rrl or rab$m_nlk;
;  2692     2764  2 !
;  2693     2765  2 ! Flag the list operation for the rights data base routines .
;  2694     2766  2 !
;  2695     2767  2 rdb_list_flag = true ;
;  2696     2768  2
;  2697     2769  2 !
;  2698     2770  2 ! Choose the appropriate display.
;  2699     2771  2 !
;  2700     2772  2
;  2701     2773  2 action = display_brief;
;  2702     2774  2 if .full_flag
;  2703     2775  2 then action = display_full;
;  2704     2776  2
;  2705     2777  2 LIB$SIGNAL(UAF$_LSTMSG1);                  ! announce starting
;  2706     2778  2
;  2707     2779  3 if rmsbad (wild_user (.action))
;  2708     2780  2 then
;  2709     2781  3     begin
;  2710     2782  3     if .rmserr eql rms$_rnf
;  2711     2783  3     then
;  2712     2784  3         LIB$SIGNAL(UAF$_BADSPC)
;  2713     2785  3     else
;  2714     2786  3         LIB$SIGNAL(UAF$_LSTERR, 0, .rmserr);
;  2715     2787  3     lstfab[fab$v_dlt] = true;            ! press delete button
;  2716     2788  3     end
;  2717     2789  2 else
;  2718     2790  2     LIB$SIGNAL(UAF$_LSTMSG2);
;  2719     2791  2
;  2720     2792  2 $disconnect (rab = lstrab);
;  2721     2793  2 $close (fab = lstfab);
;  2722     2794  1 end;
```

```
                                01FC 00000          .ENTRY  LIST_UAF, Save R2,R3,R4,R5,R6,R7,R8     ; 2687
                         58 00000000G   8F   D0 00002       MOVL    #UAF$_LSTERR, R8
                         57 00000000G   00   9E 00009       MOVAB   CLI$PRESENT, R7
                         56 00000000'   00   9E 00010       MOVAB   SD_TOKEN1, R6
                         55 00000000G   00   9E 00017       MOVAB   LIB$SIGNAL, R5
                         54 00000000'   00   9E 0001E       MOVAB   RMSERR, R4
                         53 00000000'   00   9E 00025       MOVAB   LSTRAB, R3
                                        01   DD 0002C       PUSHL   #1                              ; 2729
                                        56   DD 0002E       PUSHL   R6
                00000000G         00        02   FB 00030       CALLS   #2, PARSE_WILD
                                  01        50   E8 00037       BLBS    R0, 1$
                                            04   00003A      RET
                    FE08          C3        01   7D 0003B 1$:   MOVQ    #1, BRIEF_FLAG             ; 2736
                                      2C    A6   9F 00040       PUSHAB  SD_FULL                    ; 2738
                                  67        01   FB 00043       CALLS   #1, CLI$PRESENT
                                  09        50   E8 00046       BLBS    R0, 2$
                                      20    A6   9F 00049       PUSHAB  SD_BRIEF
                                  67        01   FB 0004C       CALLS   #1, CLI$PRESENT
                                  09        50   E8 0004F       BLBS    R0, 3$
                                FE08        C3   D4 00052 2$:   CLRL    BRIEF_FLAG                 ; 2741
                    FE0C        C3        01   D0 00056       MOVL    #1, FULL_FLAG              ; 2742
                      B5        A3          80 8F   8A 0005B 3$:   BICB2   #128, LSTFAB+5          ; 2749
                                            B0   A3 9F 00060       PUSHAB  LSTFAB                     ; 2750
                00000000G         00        01   FB 00063       CALLS   #1, SYS$CREATE
                                  64        50   D0 0006A       MOVL    R0, RMSERR
                                  0F        50   E9 0006D       BLBC    R0, 4$
                                            53   DD 00070       PUSHL   R3                         ; 2753
                00000000G         00        01   FB 00072       CALLS   #1, SYS$CONNECT
                                  64        50   D0 00079       MOVL    R0, RMSERR
                                  0A        50   E8 0007C       BLBS    R0, 5$
                                            64   DD 0007F 4$:   PUSHL   RMSERR                     ; 2754
                                            7E   D4 00081       CLRL    -(SP)
                                            58   DD 00083       PUSHL   R8
                                  65        03   FB 00085       CALLS   #3, LIB$SIGNAL
                                            04   00088       RET
                    FE10          C3        01   D0 00089 5$:   MOVL    #1, HEADER_FLAG            ; 2760
                      80          A4        63   9E 0008E       MOVAB   LSTRAB, RABPTR             ; 2761
                                      06C8  C4   D4 00092       CLRL    FOUND_MATCH                ; 2762
                    05F4          C4 00100008   8F   D0 00096       MOVL    #1048584, UAFRAB+4         ; 2763
                00000000G         00        01   90 0009F       MOVB    #1, RDB_LIST_FLAG          ; 2767
                                  52 00000000V   00   9E 000A6       MOVAB   DISPLAY_BRIEF, ACTION      ; 2773
                                  07      FE0C   C3   E9 000AD       BLBC    FULL_FLAG, 6$              ; 2774
                                  52 00000000V   00   9E 000B2       MOVAB   DISPLAY_FULL, ACTION       ; 2775
                                     00000000G   8F   DD 000B9 6$:   PUSHL   #UAF$_LSTMSG1             ; 2777
                                  65        01   FB 000BF       CALLS   #1, LIB$SIGNAL
                                            52   DD 000C2       PUSHL   ACTION                     ; 2779
                00000000V         00        01   FB 000C4       CALLS   #1, WILD_USER
                                  64        50   D0 000CB       MOVL    R0, RMSERR
                                  27        50   E8 000CE       BLBS    R0, 9$
                                  50        64   D0 000D1       MOVL    RMSERR, R0                 ; 2782
                000182B2          8F        50   D1 000D4       CMPL    R0, #98994
                                            0B   12 000DB       BNEQ    7$
                                     00000000G   8F   DD 000DD       PUSHL   #UAF$_BADSPC               ; 2784
                                  65        01   FB 000E3       CALLS   #1, LIB$SIGNAL
                                            09   11 000E6       BRB     8$
                                            50   DD 000E8 7$:   PUSHL   R0                         ; 2786
                                            7E   D4 000EA       CLRL    -(SP)
```

```
                                 58  DD 000EC          PUSHL   R8                                      ;
                                 03  FB 000EE          CALLS   #3, LIB$SIGNAL                          ;
                          65                                                                           ;
                       B5 A3          80  8F  88 000F1 8$:     BISB2   #128, LSTFAB+5                  ; 2787
                                 09  11 000F6          BRB     10$                                     ; 2779
                             00000000G   8F  DD 000F8 9$:     PUSHL   #UAF$_LSTMSG2                   ; 2790
                          65             01  FB 000FE          CALLS   #1, LIB$SIGNAL                  ;
                                 53  DD 00101 10$:     PUSHL   R3                                      ; 2792
                00000000G 00             01  FB 00103          CALLS   #1, SYS$DISCONNECT              ;
                                      B0 A3  9F 0010A          PUSHAB  LSTFAB                          ; 2793
                00000000G 00             01  FB 0010D          CALLS   #1, SYS$CLOSE                   ;
                                 04  00114          RET                                             ; 2794

; Routine Size:  277 bytes,    Routine Base:  $CODE$ + 0FFD
```

```
;  2724     2795  1 %sbttl 'show_user_uaf - display user record'
;  2725     2796  1 global routine show_user_uaf : novalue =
;  2726     2797  2 begin
;  2727     2798  2
;  2728     2799  2 !++
;  2729     2800  2 !
;  2730     2801  2 ! FUNCTIONAL DESCRIPTION:
;  2731     2802  2 !
;  2732     2803  2 !       Display the specified users on SYS$OUTPUT.
;  2733     2804  2 !
;  2734     2805  2 ! INPUTS:
;  2735     2806  2 !
;  2736     2807  2 !       none
;  2737     2808  2 !
;  2738     2809  2 ! IMPLICIT INPUTS:
;  2739     2810  2 !
;  2740     2811  2 !       none
;  2741     2812  2 !
;  2742     2813  2 ! OUTPUTS:
;  2743     2814  2 !
;  2744     2815  2 !       none
;  2745     2816  2 !
;  2746     2817  2 ! IMPLICIT OUTPUTS:
;  2747     2818  2 !
;  2748     2819  2 !       none
;  2749     2820  2 !
;  2750     2821  2 ! ROUTINE VALUE:
;  2751     2822  2 !
;  2752     2823  2 !       none
;  2753     2824  2 !
;  2754     2825  2 ! SIDE EFFECTS:
;  2755     2826  2 !
;  2756     2827  2 !       none
;  2757     2828  2 !--
;  2758     2829  2
;  2759     2830  2 local
;  2760     2831  2         action;
;  2761     2832  2
;  2762     2833  2 !
;  2763     2834  2 ! Obtain the user specification.  This sets wildcard flags and initializes
;  2764     2835  2 ! the appropriate key in RECBUF.
;  2765     2836  2 !
;  2766     2837  2
;  2767     2838  2 if not parse_wild (sd_token1,false)               ! Null string is disallowed.
;  2768     2839  2 then return;
;  2769     2840  2
;  2770     2841  2 !
;  2771     2842  2 ! Obtain qualifiers.  This determines which display should be used.
;  2772     2843  2 !
;  2773     2844  2 full_flag = true;
;  2774     2845  2 brief_flag = false;
;  2775     2846  2
;  2776     2847  3 if cli$present (sd_brief) or (not cli$present (sd_full))
;  2777     2848  2 then
;  2778     2849  3     begin
;  2779     2850  3     brief_flag = true;
;  2780     2851  3     full_flag  = false;
```

```
;  2781          2852  2      end;
;  2782          2853  2
;  2783          2854  2  !
;  2784          2855  2  ! Request a header record for the file and aim RABPTR at our RAB.
;  2785          2856  2  !
;  2786          2857  2
;  2787          2858  2  header_flag = true;
;  2788          2859  2  rabptr = outrab;
;  2789          2860  2  found_match = false;
;  2790          2861  2  uafrab[rab$l_rop] = rab$m_rrl or rab$m_nlk;
;  2791          2862  2
;  2792          2863  2  !
;  2793          2864  2  ! Flag the show operation for the rights data base routines .
;  2794          2865  2  !
;  2795          2866  2  rdb_list_flag = false ;
;  2796          2867  2
;  2797          2868  2  !
;  2798          2869  2  ! Choose the appropriate display.
;  2799          2870  2  !
;  2800          2871  2
;  2801          2872  2  action = display_full;
;  2802          2873  2  if .brief_flag
;  2803          2874  2  then action = display_brief;
;  2804          2875  2
;  2805          2876  2  if rmsbad (wild_user (.action))
;  2806          2877  2  then
;  2807          2878  3      begin
;  2808          2879  3      if .rmserr eql rms$_rnf
;  2809          2880  3      then
;  2810          2881  3          LIB$SIGNAL(UAF$_BADSPC)
;  2811          2882  3      else
;  2812          2883  3          LIB$SIGNAL(UAF$_SHOW_ERR, 0, .rmserr);
;  2813          2884  2      end;
;  2814          2885  1  end;


                                    007C 00000     .ENTRY  SHOW_USER_UAF, Save R2,R3,R4,R5,R6          ; 2796
                          56 00000000G  00  9E 00002  MOVAB   LIB$SIGNAL, R6                          ;
                          55 00000000G  00  9E 00009  MOVAB   CLI$PRESENT, R5                         ;
                          54 00000000'  00  9E 00010  MOVAB   SD_TOKEN1, R4                           ;
                          53 00000000'  00  9E 00017  MOVAB   BRIEF_FLAG, R3                          ;
                          52 00000000'  00  9E 0001E  MOVAB   RMSERR, R2                              ;
                                        7E  D4 00025  CLRL    -(SP)                                 ; 2838
                                        54  DD 00027  PUSHL   R4                                    ;
                00000000G   00          02  FB 00029  CALLS   #2, PARSE_WILD                        ;
                            7B          50  E9 00030  BLBC    R0, 5$                                ;
                      04    A3          01  D0 00033  MOVL    #1, FULL_FLAG                         ; 2844
                                        63  D4 00037  CLRL    BRIEF_FLAG                            ; 2845
                                20      A4  9F 00039  PUSHAB  SD_BRIEF                              ; 2847
                            65          01  FB 0003C  CALLS   #1, CLI$PRESENT                      ;
                            09          50  E8 0003F  BLBS    R0, 1$                                ;
                                2C      A4  9F 00042  PUSHAB  SD_FULL                               ;
                            65          01  FB 00045  CALLS   #1, CLI$PRESENT                      ;
                            03          50  E8 00048  BLBS    R0, 2$                                ;
```

```
                                63                  01   7D 0004B 1$:     MOVQ    #1, BRIEF_FLAG                  ; 2850
                             08 A3                  01   D0 0004E 2$:     MOVL    #1, HEADER_FLAG                 ; 2858
                             80 A2      0684        C2   9E 00052         MOVAB   OUTRAB, RABPTR                  ; 2859
                                        06C8        C2   D4 00058         CLRL    FOUND_MATCH                     ; 2860
                          C3F4  C2 00100008         8F   D0 0005C         MOVL    #1048584, UAFRAB+4              ; 2861
                                   00000000G        00   94 00065         CLRB    RDB_LIST_FLAG                   ; 2866
                                50 00000000V        00   9E 0006B         MOVAB   DISPLAY_FULL, ACTION            ; 2872
                                07                  63   E9 00072         BLBC    BRIEF_FLAG, 3$                  ; 2873
                                50 00000000V        00   9E 00075         MOVAB   DISPLAY_BRIEF, ACTION           ; 2874
                                                    50   DD 0007C 3$:     PUSHL   ACTION                          ; 2876
                       00000000V 00                 01   FB 0007E         CALLS   #1, WILD_USER
                                62                  50   D0 00085         MOVL    R0, RMSERR
                                23                  50   E8 00088         BLBS    R0, 5$
                                50                  62   D0 0008B         MOVL    RMSERR, R0                      ; 2879
                          000182B2 8F               50   D1 0008E         CMPL    R0, #98994
                                                    0A   12 00095         BNEQ    4$
                                   00000000G        8F   DD 00097         PUSHL   #UAF$_BADSPC                    ; 2881
                                66                  01   FB 0009D         CALLS   #1, LIB$SIGNAL
                                                    04   00A0             RET
                                                    50   DD 000A1 4$:     PUSHL   R0                              ; 2883
                                                    7E   D4 000A3         CLRL    -(SP)
                                   00000000G        8F   DD 000A5         PUSHL   #UAF$_SHOW_ERR
                                66                  03   FB 000AB         CALLS   #3, LIB$SIGNAL
                                                    04   00AE 5$:         RET                                     ; 2885

; Routine Size.  175 bytes,    Routine Base:  $CODE$ + 1112
```

```
;  2816          2886  1 %sbttl 'show_proxy - display proxy record at terminal'
;  2817          2887  1 global routine show_proxy : novalue =
;  2818          2888  2 begin
;  2819          2889  2
;  2820          2890  2 !++
;  2821          2891  2 !
;  2822          2892  2 ! FUNCTIONAL DESCRIPTION:
;  2823          2893  2 !
;  2824          2894  2 !       This routine will display a specific proxy record or
;  2825          2895  2 !       will display all proxy entries to the user terminal.
;  2826          2896  2 !
;  2827          2897  2 ! INPUTS:
;  2828          2898  2 !
;  2829          2899  2 !       none
;  2830          2900  2 !
;  2831          2901  2 ! OUTPUTS:
;  2832          2902  2 !
;  2833          2903  2 !       none
;  2834          2904  2 !
;  2835          2905  2 ! IMPLICIT INPUTS:
;  2836          2906  2 !
;  2837          2907  2 !       TOKENLEN, TOKENPTR
;  2838          2908  2 !
;  2839          2909  2 ! IMPLICIT OUTPUTS:
;  2840          2910  2 !
;  2841          2911  2 !       none
;  2842          2912  2 !
;  2843          2913  2 ! SIDE EFFECTS:
;  2844          2914  2 !
;  2845          2915  2 !       none
;  2846          2916  2 !
;  2847          2917  2 !--
;  2848          2918  2
;  2849          2919  2 local
;  2850          2920  2         node_len,
;  2851          2921  2         node_ptr,
;  2852          2922  2         remuser_len,
;  2853          2923  2         remuser_ptr,
;  2854          2924  2         action,
;  2855          2925  2         counter,
;  2856          2926  2         success;
;  2857          2927  2
;  2858          2928  2 !
;  2859          2929  2 ! Make sure that NETUAF.DAT exists
;  2860          2930  2 !
;  2861          2931  2 if not .netuaf_exists
;  2862          2932  2 then return LIB$SIGNAL(UAF$_NAFDNE);
;  2863          2933  2
;  2864          2934  2 !
;  2865          2935  2 ! Retrieve token
;  2866          2936  2 !
;  2867          2937  2 cli$get_value (sd_token1, tokendsc);
;  2868          2938  2
;  2869          2939  2 header_flag = true;
;  2870          2940  2
;  2871          2941  2 !
;  2872          2942  2 ! Wildcard spec?
```

```
;  2873          2943  2 !
;  2874          2944  2 if ch$find_ch (.tokenlen, .tokenptr, %c'%') neq 0
;  2875          2945  2 or ch$find_ch (.tokenlen, .tokenptr, %c'*') neq 0
;  2876          2946  2 then
;  2877          2947  3     begin
;  2878          2948  3     wild_netuser = true;
;  2879          2949  3     match_tokenlen = .tokenlen;
;  2880          2950  3     ch$move (.tokenlen, .tokenptr, match_token);
;  2881          2951  3     end
;  2882          2952  2 !
;  2883          2953  2 ! Otherwise, just display a single entry
;  2884          2954  2 !
;  2885          2955  2 else
;  2886          2956  3     begin
;  2887          2957  3     wild_netuser = false;
;  2888          2958  3     if not remote_parse (node_ptr, node_len, remuser_ptr, remuser_len)
;  2889          2959  3     then return;
;  2890          2960  3
;  2891          2961  3     ch$copy (.node_len, .node_ptr, ' ', naf$s_node, netbuf[naf$t_node]);
;  2892          2962  3     ch$copy (.remuser_len, .remuser_ptr, ' ',
;  2893          2963  3              naf$s_remuser, netbuf[naf$t_remuser]);
;  2894          2964  3     end;
;  2895          2965  2
;  2896          2966  2 !
;  2897          2967  2 ! Set up action routine and rab pointer
;  2898          2968  2 !
;  2899          2969  2 rabptr = outrab;
;  2900          2970  2 action = display_proxy;
;  2901          2971  2 found_match = false;
;  2902          2972  2 nafrab[rab$l_rop] = rab$m_rrl or rab$m_nlk;
;  2903          2973  2
;  2904          2974  2 !
;  2905          2975  2 ! Make call (s) necessary to display the rquested entry or entries
;  2906          2976  2 !
;  2907          2977  2 if rmsbad (locate_proxy (.action))
;  2908          2978  2 then
;  2909          2979  3     begin
;  2910          2980  3     if .rmserr eql rms$_rnf
;  2911          2981  3     then
;  2912          2982  3         LIB$SIGNAL(UAF$_BADSPC)
;  2913          2983  3     else
;  2914          2984  3         LIB$SIGNAL(UAF$_SHOW_ERR, 0, .rmserr);
;  2915          2985  2     end;
;  2916          2986  1 end;


                                00FC 00000          .ENTRY  SHOW_PROXY, Save R2,R3,R4,R5,R6,R7          ; 2887
5700000000G   00    9E 00002          MOVAB   LIB$SIGNAL, R7                              ;
5600000000'   00    9E 00009          MOVAB   TOKENDSC, R6                                ;
5E              10  C2 00010          SUBL2   #16, SP                                     ;
09            10    A6 E8 00013          BLBS    NETUAF_EXISTS, 1$                      ; 2931
   00000000G    8F  DD 00017          PUSHL   #UAF$_NAFDNE                               ; 2932
                00AA 31 0001D          BRW     7$                                         ;
                  56 DD 00020 1$:      PUSHL   R6                                         ; 2937
```

```
                                00000000'  00  9F  00022          PUSHAB  SD_TOKEN1                                    ;
                          00000000G  00    02  FB  00028          CALLS   #2, CLI$GET_VALUE                            ;
                          00000000'  00    01  D0  0002F          MOVL    #1, HEADER_FLAG                              ; 2939
                                     52    66  3C  00036          MOVZWL  TOKENLEN, R2                                 ; 2944
                                     53  04    A6  D0  00039      MOVL    TOKENPTR, R3                                 ;
                    63               52        25  3A  0003D      LOCC    #37, R2, (R3)                                ;
                                               02  12  00041      BNEQ    2$                                           ;
                                               51  D4  00043      CLRL    R1                                           ;
                                               51  D5  00045  2$:  TSTL    R1                                           ;
                                               0C  12  00047      BNEQ    4$                                           ;
                    63               52        2A  3A  00049      LOCC    #42, R2, (R3)                                ; 2945
                                               02  12  0004D      BNEQ    3$                                           ;
                                               51  D4  0004F      CLRL    R1                                           ;
                                               51  D5  00051  3$:  TSTL    R1                                           ;
                                               0F  13  00053      BEQL    5$                                           ;
                       F8    A6                01  D0  00055  4$:  MOVL    #1, WILD_NETUSER                             ; 2948
                       F4    A6                52  D0  00059      MOVL    R2, MATCH_TOKENLEN                           ; 2949
                 B0    A6         63           52  28  0005D      MOVC3   R2, (R3), MATCH_TOKEN                        ; 2950
                                               29  11  00062      BRB     6$                                           ; 2944
                                F8    A6       D4  00064  5$:  CLRL    WILD_NETUSER                                 ; 2957
                                      5E       DD  00067      PUSHL   SP                                           ; 2958
                                08    AE       9F  00069      PUSHAB  REMUSER_PTR                                  ;
                                10    AE       9F  0006C      PUSHAB  NODE_LEN                                     ;
                                18    AE       9F  0006F      PUSHAB  NODE_PTR                                     ;
                          F412  CF              04  FB  00072      CALLS   #4, REMOTE_PARSE                             ;
                                61              50  E9  00077      BLBC    R0, 9$                                       ;
          20         20   0C    BE        08    AE  2C  0007A      MOVC5   NODE_LEN, @NODE_PTR, #32, #32, NETBUF        ; 2961
                                     05A4       C6  00081                                                               ;
          20         20   04    BE             6E  2C  00084      MOVC5   REMUSER_LEN, @REMUSER_PTR, #32, #32, -       ; 2963
                                     05C4       C6  0008A                  NETBUF+32                                    ;
                          98    A6        069C  C6  9E  0008D  6$:  MOVAB   OUTRAB, RABPTR                              ; 2969
                                50  00000000V   00  9E  00093      MOVAB   DISPLAY_PROXY, ACTION                        ; 2970
                                     06E0       C6  D4  0009A      CLRL    FOUND_MATCH                                  ; 2971
                          0650  C6  00100008    8F  D0  0009E      MOVL    #1048584, NAFRAB+4                           ; 2972
                                               50  DD  000A7      PUSHL   ACTION                                       ; 2977
                    00000000V  00              01  FB  000A9      CALLS   #1, LOCATE_PROXY                             ;
                          18    A6             50  D0  000B0      MOVL    R0, RMSERR                                   ;
                                24             50  E8  000B4      BLBS    R0, 9$                                       ;
                                50        18   A6  D0  000B7      MOVL    RMSERR, R0                                   ; 2980
                    00018282  8F              50  D1  000BB      CMPL    R0, #98994                                   ;
                                               0A  12  000C2      BNEQ    8$                                           ;
                                00000000G      8F  DD  000C4      PUSHL   #UAF$_BADSPC                                 ; 2982
                                67             01  FB  000CA  7$:  CALLS   #1, LIB$SIGNAL                              ;
                                               04  000CD      RET                                                      ;
                                               50  DD  000CE  8$:  PUSHL   R0                                           ; 2984
                                               7E  D4  000D0      CLRL    -(SP)                                        ;
                                00000000G      8F  DD  000D2      PUSHL   #UAF$_SHOW_ERR                               ;
                                67             03  FB  000D8      CALLS   #3, LIB$SIGNAL                               ;
                                               04  000DB  9$:  RET                                                      ; 2986

; Routine Size:  220 bytes,    Routine Base:  $CODE$ + 11C1
```

```
;  2918     2987  1 %sbttl 'locate_proxy - access given proxy record (s)'
;  2919     2988  1 routine locate_proxy (action) =
;  2920     2989  2 begin
;  2921     2990  2
;  2922     2991  2 !++
;  2923     2992  2 !
;  2924     2993  2 ! FUNCTIONAL DESCRIPTION:
;  2925     2994  2 !
;  2926     2995  2 !       This routine will call a requested action routine a number of times.
;  2927     2996  2 !
;  2928     2997  2 ! INPUTS:
;  2929     2998  2 !
;  2930     2999  2 !       ACTION - the action routine to call for each NEUAF record
;  2931     3000  2 !
;  2932     3001  2 ! OUTPUTS:
;  2933     3002  2 !
;  2934     3003  2 !       none
;  2935     3004  2 !
;  2936     3005  2 ! SIDE EFFECTS:
;  2937     3006  2 !
;  2938     3007  2 !       none
;  2939     3008  2 !
;  2940     3009  2 !--
;  2941     3010  2
;  2942     3011  2 local
;  2943     3012  2         status;
;  2944     3013  2
;  2945     3014  2 !
;  2946     3015  2 ! If wild user, set acces to sequential and fetch all records
;  2947     3016  2 !
;  2948     3017  2 if .wild_netuser
;  2949     3018  2 then
;  2950     3019  3     begin
;  2951     3020  3     nafrab[rab$b_rac] = rab$c_seq;
;  2952     3021  3     $rewind (rab = nafrab);
;  2953     3022  3     end;
;  2954     3023  2
;  2955     3024  2 status = get_proxy_record ();
;  2956     3025  2
;  2957     3026  2 !
;  2958     3027  2 ! Fetch record and call action routine until EOF
;  2959     3028  2 !
;  2960     3029  2 if .status
;  2961     3030  2 then (.action) ();
;  2962     3031  2
;  2963     3032  2 if .wild_netuser
;  2964     3033  2 then
;  2965     3034  3     begin
;  2966     3035  3     while status = get_proxy_record ()
;  2967     3036  3     do (.action) ();
;  2968     3037  3     if .status eql rms$_eof
;  2969     3038  3     then status = true;
;  2970     3039  3     end;
;  2971     3040  2
;  2972     3041  2 !
;  2973     3042  2 ! Restore keyed access
;  2974     3043  2 !
```

```
; 2975          3044  2 nafrab[rab$b_rac] = rab$c_key;
; 2976          3045  2
; 2977          3046  2 if .wild_netuser and not .found_match
; 2978          3047  2 then LIB$SIGNAL(UAF$_BADSPC);
; 2979          3048  2
; 2980          3049  2 .status
; 2981          3050  1 end;



                                      001C 00000 LOCATE_PROXY:
                                                      .WORD    Save R2,R3,R4                          ; 2988
                    54 00000000V  00  9E 00002        MOVAB    GET_PROXY_RECORD, R4                   ;
                    53 00000000'  00  9E 00009        MOVAB    WILD_NETUSER, R3                       ;
                    0F            63  E9 00010        BLBC     WILD_NETUSER, 1$                       ; 3017
                          0672    C3  94 00013        CLRB     NAFRAB+30                              ; 3020
                          0654    C3  9F 00017        PUSHAB   NAFRAB                                 ; 3021
         00000000G  00            01  FB 0001B        CALLS    #1, SYS$REWIND                         ;
                    64            00  FB 00022 1$:    CALLS    #0, GET_PROXY_RECORD                   ; 3024
                    52            50  D0 00025        MOVL     R0, STATUS                             ;
                    04            52  E9 00028        BLBC     STATUS, 2$                             ; 3029
                04  BC            00  FB 0002B        CALLS    #0, @ACTION                            ; 3030
                    1B            63  E9 0002F 2$:    BLBC     WILD_NETUSER, 5$                       ; 3032
                    64            00  FB 00032 3$:    CALLS    #0, GET_PROXY_RECORD                   ; 3035
                    52            50  D0 00035        MOVL     R0, STATUS                             ;
                    06            52  E9 00038        BLBC     STATUS, 4$                             ;
                04  BC            00  FB 0003B        CALLS    #0, @ACTION                            ; 3036
                                  F1  11 0003F        BRB      3$                                     ;
         0001827A   8F            52  D1 00041 4$:    CMPL     STATUS, #98938                         ; 3037
                                  03  12 00048        BNEQ     5$                                     ;
                    52            01  D0 0004A        MOVL     #1, STATUS                             ; 3038
              0672  C3            01  90 0004D 5$:    MOVB     #1, NAFRAB+30                          ; 3044
                    12            63  E9 00052        BLBC     WILD_NETUSER, 6$                       ; 3046
                    0D      06E8  C3  E8 00055        BLBS     FOUND_MATCH, 6$                        ;
                       00000000G  8F  DD 0005A        PUSHL    #UAF$_BADSPC                           ; 3047
         00000000G  00            01  FB 00060        CALLS    #1, LIB$SIGNAL                         ;
                    50            52  D0 00067 6$:    MOVL     STATUS, R0                             ; 3050
                                      04 0006A        RET                                             ;

; Routine Size:  107 bytes,    Routine Base:  $CODE$ + 129D
```

```
;  2983          3051  1 %sbttl 'get_proxy_record - read single proxy record'
;  2984          3052  1 routine get_proxy_record =
;  2985          3053  2 begin
;  2986          3054  2
;  2987          3055  2 !++
;  2988          3056  2 !
;  2989          3057  2 ! FUNCTIONAL DESCRIPTION:
;  2990          3058  2 !
;  2991          3059  2 !       This routine accesses a specific NETUAF.DAT record
;  2992          3060  2 !
;  2993          3061  2 ! INPUTS:
;  2994          3062  2 !
;  2995          3063  2 !       none
;  2996          3064  2 !
;  2997          3065  2 ! OUTPUTS:
;  2998          3066  2 !
;  2999          3067  2 !       none
;  3000          3068  2 !
;  3001          3069  2 ! SIDE EFFECTS:
;  3002          3070  2 !
;  3003          3071  2 !       none
;  3004          3072  2 !
;  3005          3073  2 !--
;  3006          3074  2
;  3007          3075  2 local
;  3008          3076  2         counter,
;  3009          3077  2         success;
;  3010          3078  2
;  3011          3079  2 counter = retry_rlk;
;  3012          3080  2
;  3013          3081  3 while ((success = $get (rab = nafrab)) eql rms$_rlk)
;  3014          3082  3       and  ((counter = .counter - 1) geq 0);
;  3015          3083  2 do
;  3016          3084  2    if $schdwk (daytim = wakedelta) then $hiber;
;  3017          3085  2
;  3018          3086  2 .success
;  3019          3087  1 end;


                                                .EXTRN  SYS$GET, SYS$SCHDWK
                                                .EXTRN  SYS$HIBER

                                        000C 00000 GET_PROXY_RECORD:
                                                .WORD   Save R2,R3                      ;  3052
                    52                  08   D0 00002          MOVL    #8, COUNTER     ;  3079
                            00000000'   00   9F 00005 1$:      PUSHAB  NAFRAB          ;  3081
00000000G   00                          01   FB 0000B          CALLS   #1, SYS$GET
                    53                  50   D0 00012          MOVL    R0, SUCCESS
000182AA    8F                          53   D1 00015          CMPL    SUCCESS, #98986
                                        21   12 0001C          BNEQ    2$
                                        52   D7 0001E          DECL    COUNTER         ;  3082
                                        1D   19 00020          BLSS    2$
                                        7E   D4 00022          CLRL    -(SP)           ;  3084
                            00000000'   00   9F 00024          PUSHAB  WAKEDELTA
                                        7E   7C 0002A          CLRQ    -(SP)
00000000G   00                          04   FB 0002C          CALLS   #4, SYS$SCHDWK  ;
```

```
                         CF              50  E9 00033          BLBC    R0, 1$                                  ;
                00000000G 00             00  FB 00036          CALLS   #0, SYS$HIBER                           ;
                                         C6  11 0003D          BRB     1$                                      ;
                         50              53  D0 0003F 2$:      MOVL    SUCCESS, R0                             ; 3087
                                             04 00042          RET                                             ;

; Routine Size:  67 bytes,    Routine Base:  $CODE$ + 1308
```

```
;  3021          3088  1 %sbttl 'display_proxy - format and output a proxy entry'
;  3022          3089  1 routine display_proxy : novalue =
;  3023          3090  2 begin
;  3024          3091  2
;  3025          3092  2 !++
;  3026          3093  2 !
;  3027          3094  2 ! FUNCTIONAL DESCRIPTION:
;  3028          3095  2 !
;  3029          3096  2 !       This routine formats and outputs a line of a NETUAF.DAT entry
;  3030          3097  2 !
;  3031          3098  2 ! INPUTS:
;  3032          3099  2 !
;  3033          3100  2 !       none
;  3034          3101  2 !
;  3035          3102  2 ! OUTPUTS:
;  3036          3103  2 !
;  3037          3104  2 !       none
;  3038          3105  2 !
;  3039          3106  2 ! SIDE EFFECTS:
;  3040          3107  2 !
;  3041          3108  2 !       none
;  3042          3109  2 !
;  3043          3110  2 !--
;  3044          3111  2
;  3045          3112  2 bind
;  3046          3113  2         nafhdr          = cstring (' Node    Remote User      Local User'),
;  3047          3114  2         shownaf         = cstring ('!6AD::!12AD    !12AD');
;  3048          3115  2
;  3049          3116  2
;  3050          3117  2 if .wild_netuser
;  3051          3118  2 and not
;  3052          3119  3     begin
;  3053          3120  3     local
;  3054          3121  3         nodelen, usrlen, proxy_buf : vector[naf$s_remname+2,byte];
;  3055          3122  3     map
;  3056          3123  3         dbl_colon : vector;
;  3057          3124  3
;  3058          3125  3     nodelen = namelen (naf$s_node, netbuf[naf$t_node]);
;  3059          3126  3     usrlen  = namelen (naf$s_remuser, netbuf[naf$t_remuser]);
;  3060          3127  3     ch$move (.nodelen, netbuf[naf$t_node], proxy_buf [0]);
;  3061          3128  3     ch$move (2, .dbl_colon[1], proxy_buf [.nodelen]);
;  3062          3129  3     ch$move (.usrlen, netbuf [naf$t_remuser], proxy_buf [.nodelen+2]);
;  3063          3130  3
;  3064          3131  3     usrlen = .nodelen + .usrlen + 2;
;  3065          3132  3     fmg$match_name (.usrlen, proxy_buf, .match_tokenlen, match_token)
;  3066          3133  3     end
;  3067          3134  2 then
;  3068          3135  2     return;
;  3069          3136  2
;  3070          3137  2 found_match = true;
;  3071          3138  2
;  3072          3139  2 if .header_flag
;  3073          3140  2 then
;  3074          3141  3     begin
;  3075          3142  3     faomac (nafhdr);
;  3076          3143  3     output_null;
;  3077          3144  3     header_flag = false;
```

```
;  3078          3145  2     end;
;  3079          3146  2
;  3080        P 3147  2 faomac (shownaf,
;  3081        P 3148  2 !***     naf$s_node,      netbuf[naf$t_node],
;  3082        P 3149  2 !***     naf$s_remuser,   netbuf[naf$t_remuser],
;  3083        P 3150  2 !***     naf$s_localuser, netbuf[naf$t_localuser]);
;  3084        P 3151  2         6, netbuf[naf$t_node],
;  3085        P 3152  2         12, netbuf[naf$t_remuser],
;  3086          3153  2         12, netbuf[naf$t_localuser]);
;  3087          3154  2
;  3088          3155  1 end;


                                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                                 22 0026C P.ACQ:  .BYTE   34
20 65 74 6F 6D 65 52 20 20 20 65 64 6F 4E 20 0026D                       .ASCII  \ Node   Remote User     Local User\
20 6C 61 63 6F 4C 20 20 20 20 20 72 65 73 55 0027C                                                                    ;
                                 72 65 73 55 0028B                                                                    ;
                                                                 14 0028F P.ACR:  .BYTE   20                          ;
20 20 20 20 44 41 32 31 21 3A 3A 44 41 36 21 00290                       .ASCII  \!6AD::!12AD     !12AD\              ;
                                 44 41 32 31 21 0029F                                                                 ;

                                                                 NAFHDR=          P.ACQ                               ;
                                                                 SHOWNAF=         P.ACR
                                                                        .EXTRN  SYS$FAO

                                                                        .PSECT  $CODE$,NOWRT,2

                                                          OFFC 00000 DISPLAY_PROXY:
                                                                        .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11    ; 3089
                                                    5B 00000000'   00   9E 00002        MOVAB   HEADER_FLAG, R11            ;
                                                    5A 00000000G   00   9E 00009        MOVAB   SYS$PUT, R10                ;
                                                    59 00000000'   00   9E 00010        MOVAB   DBL_COLON+4, R9             ;
                                                    58 00000000'   00   9E 00017        MOVAB   RABPTR, R8                  ;
                                                    5E        BC   AE   9E 0001E        MOVAB   -68(SP), SP                 ;
                                                    47        60   A8   E9 00022        BLBC    WILD_NETUSER, 1$            ; 3117
                                          060C   C8              20   3A 00026        LOCC    #32, #32, NETBUF            ; 3125
                                                 56               20   C3 0002C        SUBL3   R0, #32, NODELEN            ;
                                          062C   C8              20   3A 00030        LOCC    #32, #32, NETBUF+32         ; 3126
                                                 57               20   C3 00036        SUBL3   R0, #32, USRLEN             ;
                                                 6E      060C    C8   28 0003A        MOVC3   NODELEN, NETBUF, PROXY_BUF  ; 3127
                                                    50           69   D0 00040        MOVL    DBL_COLON+4, R0             ; 3128
                                                              6E46   9F 00043        PUSHAB  PROXY_BUF[NODELEN]          ;
                                                    9E            60   B0 00046        MOVW    (R0), @(SP)+                ;
                                             02 AE46      062C    C8   28 00049        MOVC3   USRLEN, NETBUF+32, PROXY_BUF+2[NODELEN] ; 3129
                                                    57        02 A746   9E 00051        MOVAB   2(USRLEN)[NODELEN], USRLEN  ; 3131
                                                    55        18   A8   9E 00056        MOVAB   MATCH_TOKEN, R5             ; 3132
                                                    53             6E   9E 0005A        MOVAB   PROXY_BUF, R3               ;
                                                    54        5C   A8   D0 0005D        MOVL    MATCH_TOKENLEN, R4          ;
                                                    52             57   D0 00061        MOVL    USRLEN, R2                  ;
                                                       00000000G   00   16 00064        JSB     FMG$MATCH_NAME              ;
                                                    6B             50   E9 0006A        BLBC    R0, 3$                      ;
                                                0748   C8              01   D0 0006D 1$:  MOVL    #1, FOUND_MATCH             ; 3137
                                                    2F              6B   E9 00072        BLBC    HEADER_FLAG, 2$             ; 3139
                                                    F8   A8    0158   C9   9B 00075        MOVZBW  NAFHDR, FAODSC              ; 3142
```

```
                    FC   A8   0159  C9  9E 0007B        MOVAB   NAFHDR+1, FAODSC+4                ;
                         F0   A8  9F 00081              PUSHAB  DISDSC                           ;
                7E            68  22  C1 00084        ADDL3   #34, RABPTR, -(SP)               ;
                              F8  A8  9F 00088          PUSHAB  FAODSC                           ;
             00000000G        00  03  FB 0008B          CALLS   #3, SYS$FAO                      ;
                                  68  DD 00092          PUSHL   RABPTR                           ;
                              6A  01  FB 00094          CALLS   #1, SYS$PUT                      ;
                              50  68  DO 00097          MOVL    RABPTR, RO                       ;
                                  22  A0 B4 0009A      CLRW    34(RO)                           ;
                                  50  DD 0009D          PUSHL   RO                               ;
                              6A  01  FB 0009F          CALLS   #1, SYS$PUT                      ;
                                  6B  D4 000A2          CLRL    HEADER_FLAG                      ; 3144
                    F8   A8   017B  C9  9B 000A4 2$:    MOVZBW  SHOWNAF, FAODSC                  ; 3153
                    FC   A8   017C  C9  9E 000AA        MOVAB   SHOWNAF+1, FAODSC+4              ;
                              064C  C8  9F 000B0        PUSHAB  NETBUF+64                        ;
                                    0C  DD 000B4        PUSHL   #12                              ;
                              062C  C8  9F 000B6        PUSHAB  NETBUF+32                        ;
                                    0C  DD 000BA        PUSHL   #12                              ;
                              060C  C8  9F 000BC        PUSHAB  NETBUF                           ;
                                    06  DD 000C0        PUSHL   #6                               ;
                                  F0  A8  9F 000C2      PUSHAB  DISDSC                           ;
                7E            68  22  C1 000C5        ADDL3   #34, RABPTR, -(SP)               ;
                              F8  A8  9F 000C9          PUSHAB  FAODSC                           ;
             00000000G        00  09  FB 000CC          CALLS   #9, SYS$FAO                      ;
                                  68  DD 000D3          PUSHL   RABPTR                           ;
                              6A  01  FB 000D5          CALLS   #1, SYS$PUT                      ;
                                  04  000D8 3$:        RET                                      ; 3155

; Routine Size:  217 bytes,    Routine Base:  $CODE$ + 134B
```

```
;  3090     3156  1 %sbttl 'wild_user - user wild card routine'
;  3091     3157  1 global routine wild_user (action) =
;  3092     3158  2 begin
;  3093     3159  2
;  3094     3160  2 !++
;  3095     3161  2 !
;  3096     3162  2 ! FUNCTIONAL DESCRIPTION:
;  3097     3163  2 !
;  3098     3164  2 !       Provide a general means of accessing the User Authorization File
;  3099     3165  2 !       records.  There are six methods:
;  3100     3166  2 !
;  3101     3167  2 !                       UGMS
;  3102     3168  2 !                       IRET
;  3103     3169  2 !                       CPMR
;  3104     3170  2 !                       \\\\
;  3105     3171  2 !                       FWWW
;  3106     3172  2 !                       LIII
;  3107     3173  2 !                       ALLL
;  3108     3174  2 !       Syntax          GDDD    Interpretation
;  3109     3175  2 !       ------          ----    --------------
;  3110     3176  2 !       Username        FFFF    Exactly one user is to be located
;  3111     3177  2 !       *               FFFT    All users (alphabetically)
;  3112     3178  2 !       [Group,Member]  TFFF    All users with the specified UIC
;  3113     3179  2 !       [Group,*]       TFTF    All users in the specified group (by member)
;  3114     3180  2 !       [*,Member]      TTFF    A FIFO listing of the groups with this member
;  3115     3181  2 !       [*,*]           TTTF    All users by UIC
;  3116     3182  2 !
;  3117     3183  2 ! INPUTS:
;  3118     3184  2 !
;  3119     3185  2 !       ACTION - Pointer to routine to call after each successful GET
;  3120     3186  2 !
;  3121     3187  2 ! IMPLICIT INPUTS:
;  3122     3188  2 !
;  3123     3189  2 !       UIC_FLAG - UIC form (instead of username)
;  3124     3190  2 !       GRP_WILD - Group wild card (must imply UIC_FLAG)
;  3125     3191  2 !       MEM_WILD - Member wild card (must imply UIC_FLAG)
;  3126     3192  2 !       STR_WILD - all users alphabetically (must imply NOT UIC_FLAG)
;  3127     3193  2 !       UAFRAB - RMS data structure for SYSUAF.DAT
;  3128     3194  2 !       RECBUF - The current record
;  3129     3195  2 !
;  3130     3196  2 ! OUTPUTS:
;  3131     3197  2 !
;  3132     3198  2 !       none
;  3133     3199  2 !
;  3134     3200  2 ! IMPLICIT OUTPUTS:
;  3135     3201  2 !
;  3136     3202  2 !       none
;  3137     3203  2 !
;  3138     3204  2 ! ROUTINE VALUE:
;  3139     3205  2 !
;  3140     3206  2 !       If an abnormal condition is encountered the appropriate status
;  3141     3207  2 !       is returned.
;  3142     3208  2 !
;  3143     3209  2 ! SIDE EFFECTS:
;  3144     3210  2 !
;  3145     3211  2 !       none
;  3146     3212  2 !--
```

```
;  3147     3213  2
;  3148     3214  2 macro
;  3149     3215  2         lmt_l_uic = 0,0,32,0%,          ! User ID Code
;  3150     3216  2         lmt_w_mem = 0,0,16,0%,          ! Member subfield
;  3151     3217  2         lmt_w_grp = 2,0,16,0%;          ! Group subfield
;  3152     3218  2
;  3153     3219  2 local
;  3154     3220  2         status,                         ! This routine's status
;  3155     3221  2         lmtkey  : block[4,byte];        ! Limiting key value for sequential loop
;  3156     3222  2
;  3157     3223  2 if .uic_flag
;  3158     3224  2 then
;  3159     3225  2 !
;  3160     3226  2 ! Change the key of reference and the key buffer if a UIC form was
;  3161     3227  2 ! specified.
;  3162     3228  2 !
;  3163     3229  2     begin
;  3164     3230  2     uafrab[rab$b_krf] = 1;
;  3165     3231  2     uafrab[rab$l_kbf] = recbuf[uaf$l_uic];
;  3166     3232  2     uafrab[rab$b_ksz] = 4;
;  3167     3233  2     end;
;  3168     3234  2
;  3169     3235  2 if .mem_wild and not .grp_wild
;  3170     3236  2 then
;  3171     3237  2 !
;  3172     3238  2 ! The UIC requested was of the form [Group,*]
;  3173     3239  2 !
;  3174     3240  2     uafrab[rab$v_kge] = true;
;  3175     3241  2
;  3176     3242  2 !
;  3177     3243  2 ! LMTKEY need be loaded only IF .UIC_FLAG AND NOT (.GRP_WILD AND .MEM_WILD)
;  3178     3244  2 ! but it is simpler to always load it.
;  3179     3245  2 !
;  3180     3246  2
;  3181     3247  2 lmtkey[lmt_l_uic] = .recbuf[uaf$l_uic];
;  3182     3248  2
;  3183     3249  2 !
;  3184     3250  2 ! Locate the first user meeting the specification.
;  3185     3251  2 !
;  3186     3252  2
;  3187     3253  2 if .str_wild or .grp_wild
;  3188     3254  2 then
;  3189     3255  2     begin
;  3190     3256  2 !
;  3191     3257  2 ! Every user in the file is to be accessed.
;  3192     3258  2 !
;  3193     3259  2     uafrab[rab$b_rac] = rab$c_seq;
;  3194     3260  2     $rewind (rab = uafrab);
;  3195     3261  2     status = get_uaf_record ();
;  3196     3262  2     end
;  3197     3263  2 else
;  3198     3264  2     begin
;  3199     3265  2     status = get_uaf_record ();
;  3200     3266  2     if .uic_flag
;  3201     3267  2     then
;  3202     3268  4         begin
;  3203     3269  4 !
```

```
;  3204     3270  4 ! Even an explicit UIC requires sequential reads to locate duplicates.
;  3205     3271  4 !
;  3206     3272  4         uafrab[rab$b_rac] = rab$c_seq;
;  3207     3273  4         if .mem_wild and not .grp_wild
;  3208     3274  4         then
;  3209     3275  5             begin
;  3210     3276  5 !
;  3211     3277  5 ! RAB$V_KGE is set on the initial access for specifications of the
;  3212     3278  5 ! form [Group,*] so if the specified group has no members the record
;  3213     3279  5 ! will be that of a user in ancther group.
;  3214     3280  5 !
;  3215     3281  5             uafrab[rab$v_kge] = false;
;  3216     3282  5             if .lmtkey[lmt_w_grp] nequ .recbuf[uaf$w_grp]
;  3217     3283  5             then
;  3218     3284  5                 status = rms$_rnf;
;  3219     3285  4             end;
;  3220     3286  3         end;
;  3221     3287  2     end;
;  3222     3288  2
;  3223     3289  2 if .status
;  3224     3290  2 then
;  3225     3291  3     begin
;  3226     3292  3 !
;  3227     3293  3 ! Feed the action routine the first record.  In the case of an explicit
;  3228     3294  3 ! username specification this will be the only record.
;  3229     3295  3 !
;  3230     3296  3     while .status
;  3231     3297  3     do
;  3232     3298  4         begin
;  3233     3299  5         if (
;  3234     3300  5             if .grp_wild and not .mem_wild
;  3235     3301  5             then .recbuf[uaf$w_mem] eql .lmtkey[lmt_w_mem]
;  3236     3302  5             else true
;  3237     3303  5 )
;  3238     3304  4         then status = (.action) ();
;  3239     3305  4         if not .status then exitloop;
;  3240     3306  5         if not (.str_wild or .uic_flag)
;  3241     3307  4         then exitloop;
;  3242     3308  4         status = get_uaf_record ();
;  3243     3309  4         if not .status then exitloop;
;  3244     3310  4         if .uic_flag
;  3245     3311  4         then
;  3246     3312  5             begin
;  3247     3313  5 !
;  3248     3314  5 ! The limiting key value is used in different ways depending
;  3249     3315  5 ! on the form of the UIC specification.
;  3250     3316  5 !
;  3251     3317  5             if .mem_wild and not .grp_wild
;  3252     3318  5             then
;  3253     3319  5 !
;  3254     3320  5 ! [Group,*]
;  3255     3321  5 !
;  3256     3322  6                 begin
;  3257     3323  6                 if .lmtkey[lmt_w_grp] nequ .recbuf[uaf$w_grp]
;  3258     3324  6                 then exitloop;
;  3259     3325  5                 end;
;  3260     3326  6             if not (.grp_wild or .mem_wild)
```

```
;  3261     3327  5                 then
;  3262     3328  5 !
;  3263     3329  5 ! [Group,Member]
;  3264     3330  5 !
;  3265     3331  6                     begin
;  3266     3332  6                     if .lmtkey[lmt_l_uic] nequ .recbuf[uaf$l_uic]
;  3267     3333  6                     then exitloop;
;  3268     3334  5                     end;
;  3269     3335  4                 end;
;  3270     3336  3             end;                            ! end of wild carding loop
;  3271     3337  3         if .status eql rms$_eof
;  3272     3338  3         then status = true;                 ! Hitting EOF is ok
;  3273     3339  2         end;
;  3274     3340  2
;  3275     3341  2 if .str_wild and not .found_match
;  3276     3342  2 then LIB$SIGNAL(UAF$_BADSPC);
;  3277     3343  2
;  3278     3344  2 !
;  3279     3345  2 ! The RAB must be returned to its former state before exiting.
;  3280     3346  2 !
;  3281     3347  2 uafrab[rab$b_rac] = rab$c_key;
;  3282     3348  2
;  3283     3349  2 if .uic_flag
;  3284     3350  2 then
;  3285     3351  3     begin
;  3286     3352  3     uafrab[rab$b_krf] = 0;
;  3287     3353  3     uafrab[rab$l_kbf] = recbuf[uaf$t_username];
;  3288     3354  3     uafrab[rab$b_ksz] = uaf$s_username;
;  3289     3355  2     end;
;  3290     3356  2
;  3291     3357  2 !
;  3292     3358  2 ! Reset parse count
;  3293     3359  2 !
;  3294     3360  2 call_count = 0;
;  3295     3361  2
;  3296     3362  2 .status
;  3297     3363  1 end;


                                                001C 00000         .ENTRY  WILD_USER, Save R2,R3,R4                        ; 3157
                                    54 00000000V    00    9E 00002 MOVAB   GET_UAF_RECORD, R4                              ;
                                    53 00000000'    00    9E 00009 MOVAB   GRP_WILD, R3                                    ;
                                    5E              04    C2 00010 SUBL2   #4, SP                                          ;
                                    0E        FC    A3    E9 00013 BLBC    UIC_FLAG, 1$                                    ; 3223
                                FF50  C3    F95C    C3    9E 00017 MOVAB   RECBUF+36, UAFRAB+48                            ; 3231
                                FF54  C3    0104    8F    B0 0001E MOVW    #260, UAFRAB+52                                 ; 3232
                                    08      04    A3    E9 00025 1$:     BLBC    MEM_WILD, 2$                            ; 3235
                                    05            63    E8 00029         BLBS    GRP_WILD, 2$                            ;
                                FF26  C3          20    88 0002C         BISB2   #32, UAFRAB+6                           ; 3240
                                    6E    F95C    C3    D0 00031 2$:     MOVL    RECBUF+36, LMTKEY                       ; 3247
                                    03      08    A3    E8 00036         BLBS    STR_WILD, 3$                            ; 3253
                                    17            63    E9 0003A         BLBC    GRP_WILD, 4$                            ;
                                          FF3E    C3    94 0003D 3$:     CLRB    UAFRAB+30                               ; 3259
                                          FF20    C3    9F 00041         PUSHAB  UAFRAB                                  ; 3260
```

```
                00000000G  00              01 FB 00045          CALLS   #1, SYS$REWIND                          ;
                           64              00 FB 0004C          CALLS   #0, GET_UAF_RECORD                     ; 3261
                           52              50 D0 0004F          MOVL    R0, STATUS                             ;
                                           29 11 00052          BRB     5$                                     ; 3253
                           64              00 FB 00054 4$:      CALLS   #0, GET_UAF_RECORD                     ; 3265
                           52              50 D0 00057          MOVL    R0, STATUS                             ;
                           1F          FC  A3 E9 0005A          BLBC    UIC_FLAG, 5$                           ; 3266
                       FF3E            C3 94 0005E          CLRB    UAFRAB+30                              ; 3272
                           17          04  A3 E9 00062          BLBC    MEM_WILD, 5$                           ; 3273
                           14              63 E8 00066          BLBS    GRP_WILD, 5$                           ;
                     FF26  C3              20 8A 00069          BICB2   #32, UAFRAB+6                          ; 3281
                     F95E  C3          02  AE B1 0006E          CMPW    LMTKEY+2, RECBUF+38                    ; 3282
                                           07 13 00074          BEQL    5$                                     ;
                           52 000182B2     8F D0 00076          MOVL    #98994, STATUS                         ; 3284
                           5B              52 E9 0007D 5$:      BLBC    STATUS, 12$                            ; 3289
                           4C              52 E9 00080 6$:      BLBC    STATUS, 11$                            ; 3296
                           0B              63 E9 00083          BLBC    GRP_WILD, 7$                           ; 3300
                           07          04  A3 E8 00086          BLBS    MEM_WILD, 7$                           ;
                           6E      F95C    C3 B1 0008A          CMPW    RECBUF+36, LMTKEY                      ; 3301
                                           07 12 0008F          BNEQ    8$                                     ;
                       04  BC              00 FB 00091 7$:      CALLS   #0, @ACTION                            ; 3304
                           52              50 D0 00095          MOVL    R0, STATUS                             ;
                           34              52 E9 00098 8$:      BLBC    STATUS, 11$                            ; 3305
                           04          08  A3 E8 0009B          BLBS    STR_WILD, 9$                           ; 3306
                           2C          FC  A3 E9 0009F          BLBC    UIC_FLAG, 11$                          ;
                           64              00 FB 000A3 9$:      CALLS   #0, GET_UAF_RECORD                     ; 3308
                           52              50 D0 000A6          MOVL    R0, STATUS                             ;
                           23              52 E9 000A9          BLBC    STATUS, 11$                            ; 3309
                           D0          FC  A3 E9 000AC          BLBC    UIC_FLAG, 6$                           ; 3310
                           50          04  A3 D0 000B0          MOVL    MEM_WILD, R0                           ; 3317
                           0B              50 E9 000B4          BLBC    R0, 10$                                ;
                           C6              63 E8 000B7          BLBS    GRP_WILD, 6$                           ;
                     F95E  C3          02  AE B1 000BA          CMPW    LMTKEY+2, RECBUF+38                    ; 3323
                                           0D 12 000C0          BNEQ    11$                                    ;
                           BB              63 E8 000C2 10$:     BLBS    GRP_WILD, 6$                           ; 3326
                           B8              50 E8 000C5          BLBS    R0, 6$                                 ;
                     F95C  C3              6E D1 000C8          CMPL    LMTKEY, RECBUF+36                      ; 3332
                                           B1 13 000CD          BEQL    6$                                     ;
                0001827A  8F              52 D1 000CF 11$:     CMPL    STATUS, #98938                         ; 3337
                                           03 12 000D6          BNEQ    12$                                    ;
                           52              01 D0 000D8          MOVL    #1, STATUS                             ; 3338
                           11          08  A3 E9 000DB 12$:     BLBC    STR_WILD, 13$                          ; 3341
                           0D          F8  A3 E8 000DF          BLBS    FOUND_MATCH, 13$                       ;
                                   00000000G   8F DD 000E3          PUSHL   #UAF$_BADSPC                           ; 3342
                00000000G  00              01 FB 000E9          CALLS   #1, LIB$SIGNAL                         ;
                     FF3E  C3              01 90 000F0 13$:     MOVB    #1, UAFRAB+30                          ; 3347
                           0C          FC  A3 E9 000F5          BLBC    UIC_FLAG, 14$                          ; 3349
                     FF50  C3      F93C    C3 9E 000F9          MOVAB   RECBUF+4, UAFRAB+48                    ; 3353
                     FF54  C3              20 B0 00100          MOVW    #32, UAFRAB+52                         ; 3354
                                   F914    C3 D4 00105 14$:     CLRL    CALL_COUNT                             ; 3360
                           50              52 D0 00109          MOVL    STATUS, R0                             ; 3363
                                           04 0010C             RET                                            ;

; Routine Size:  269 bytes,    Routine Base:  $CODE$ + 1424
```

```
;  3299     3364  1 %sbttl 'display_brief - writes a brief user display'
;  3300     3365  1 routine display_brief =
;  3301     3366  2 begin
;  3302     3367  2
;  3303     3368  2 !++
;  3304     3369  2 !
;  3305     3370  2 ! FUNCTIONAL DESCRIPTION:
;  3306     3371  2 !
;  3307     3372  2 !       Provide an ASCII listing of the most important record information
;  3308     3373  2 !       (username, owner, etc.) for each record supplied.
;  3309     3374  2 !
;  3310     3375  2 ! INPUTS:
;  3311     3376  2 !
;  3312     3377  2 !       none
;  3313     3378  2 !
;  3314     3379  2 ! IMPLICIT INPUTS:
;  3315     3380  2 !
;  3316     3381  2 !       RABPTR - RMS data structure for the file
;  3317     3382  2 !
;  3318     3383  2 ! OUTPUTS:
;  3319     3384  2 !
;  3320     3385  2 !       none
;  3321     3386  2 !
;  3322     3387  2 ! IMPLICIT OUTPUTS:
;  3323     3388  2 !
;  3324     3389  2 !       none
;  3325     3390  2 !
;  3326     3391  2 ! ROUTINE VALUE:
;  3327     3392  2 !
;  3328     3393  2 !       none
;  3329     3394  2 !
;  3330     3395  2 ! SIDE EFFECTS:
;  3331     3396  2 !
;  3332     3397  2 !       none
;  3333     3398  2 !--
;  3334     3399  2
;  3335     3400  2 bind
;  3336   P 3401  2 lststr1 = cstring ('       Owner         Username           UIC       Account  Privs',
;  3337     3402  2                   ' Pri Directory'),
;  3338     3403  2 lststr2 = cstring ('!20AC !12AD !15%U !8AF !6AC !2UL !AC!AC');
;  3339     3404  2
;  3340     3405  2 !
;  3341     3406  2 ! Output a header if one was requested.
;  3342     3407  2 !
;  3343     3408  2 if .header_flag
;  3344     3409  2 then
;  3345     3410  3     begin
;  3346     3411  3     faomac (lststr1);
;  3347     3412  3     output_null;
;  3348     3413  3     header_flag = false;
;  3349     3414  2     end;
;  3350     3415  2
;  3351     3416  2 if .str_wild and not fmg$match_name (namelen (uaf$s_username,recbuf[uaf$t_username]),
;  3352     3417  2                                        recbuf[uaf$t_username],
;  3353     3418  3                                        .match_tokenlen,  match_token)
;  3354     3419  2 then return true;
;  3355     3420  2
```

```
;  3356          3421  2 found_match = true;
;  3357          3422  2
;  3358          3423  2 !
;  3359          3424  2 ! Output the record.
;  3360          3425  2 !
;  3361          3426  2
;  3362          3427  2 ch$fill (' ', disbuflen, disbuf);
;  3363          3428  2
;  3364        P 3429  2 faomac (lststr2,
;  3365        P 3430  2     recbuf[uaf$t_owner],
;  3366        P 3431  2 !***     uaf$s_username, recbuf[uaf$t_username],
;  3367        P 3432  2     12, recbuf[uaf$t_username],
;  3368        P 3433  2     .recbuf[uaf$l_uic],
;  3369        P 3434  2 !***     uaf$s_account, recbuf[uaf$t_account],
;  3370        P 3435  2     8, recbuf[uaf$t_account],
;  3371        P 3436  2     classify_priv (recbuf[uaf$q_priv], .recbuf[uaf$l_uic]),
;  3372        P 3437  2     .recbuf[uaf$b_pri],
;  3373        P 3438  2     recbuf[uaf$t_defdev],
;  3374          3439  2     recbuf[uaf$t_defdir]);
;  3375          3440  2
;  3376          3441  2 true
;  3377          3442  1 end;


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                 4E  002A4 P.ACS:  .BYTE   78
20  20  20  72  65  6E  77  4F  20  20  20  20  20  20  20  002A5        .ASCII  \       Owner           Username        \
20  65  6D  61  6E  72  65  73  55  20  20  20  20  20  20  002B4
                    20  20  20  20  20  20  20  20  20  20  002C3
75  6F  63  63  41  20  20  20  20  20  20  20  43  49  55  002CD        .ASCII  \UIC        Account  Privs Pri Directory\
44  20  69  72  50  20  73  76  69  72  50  20  20  74  6E  002DC
                        79  72  6F  74  63  65  72  69  6E  002EB
                                                 27  002F3 P.ACT:  .BYTE   39
35  31  21  20  44  41  32  31  21  20  43  41  30  32  21  002F4        .ASCII  \!20AC !12AD !15%U !8AF !6AC !2UL !AC!AC\
32  21  20  43  41  36  21  20  46  41  38  21  20  55  25  00303
                    43  41  21  43  41  21  20  4C  55  00312

                                                LSTSTR1=            P.ACS
                                                LSTSTR2=            P.ACT

                                                        .PSECT  $CODE$,NOWRT,2

                                                 07FC 00000 DISPLAY_BRIEF:
                                                        .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10                ; 3365
                                  5A  00000000'  00  9E  00002          MOVAB   HEADER_FLAG, R10
                                  59  00000000G  00  9E  00009          MOVAB   SYS$FAO, R9
                                  58  00000000G  00  9E  00010          MOVAB   SYS$PUT, R8
                                  57  00000000'  00  9E  00017          MOVAB   LSTSTR1, R7
                                  56  00000000'  00  9E  0001E          MOVAB   RABPTR, R6
                                  28             6A  E9  00025          BLBC    HEADER_FLAG, 1$                 ; 3408
                              F8  A6             67  9B  00028          MOVZBW  LSTSTR1, FAODSC                 ; 3411
                              FC  A6         01  A7  9E  0002C          MOVAB   LSTSTR1+1, FAODSC+4
                                             F0  A6  9F  00031          PUSHAB  DISDSC
                          7E                 66  22  C1  00034          ADDL3   #34, RABPTR, -(SP)
```

```
                                                F8  A6  9F  00038          PUSHAB  FAODSC                                  ;
                                            69      03  FB  0003B          CALLS   #3, SYS$FAO                             ;
                                                    66  DD  0003E          PUSHL   RABPTR                                  ;
                                            68          01  FB  00040      CALLS   #1, SYS$PUT                             ;
                                            50          66  D0  00043      MOVL    RABPTR, R0                              ;
                                                    22  A0  B4  00046      CLRW    34(R0)                                  ;
                                                        50  DD  00049      PUSHL   R0                                      ;
                                            68          01  FB  0004B      CALLS   #1, SYS$PUT                             ;
                                                        6A  D4  0004E      CLRL    HEADER_FLAG                             ; 3413
                                            23      0758    C6  E9  00050 1$:  BLBC    STR_WILD, 2$                        ; 3416
                                            55        18    A6  9E  00055      MOVAB   MATCH_TOKEN, R5                         ; 3417
                                            53      008C    C6  9E  00059      MOVAB   RECBUF+4, R3                            ;
                                    008C    C6      20          20  3A  0005E  LOCC    #32, #32, RECBUF+4                  ; 3416
                                            52        E0    A0  9E  00064      MOVAB   -32(R0), R2                             ;
                                            52              52  CE  00068      MNEGL   R2, R2                                  ;
                                            54        5C    A6  D0  0006B      MOVL    MATCH_TOKENLEN, R4                      ; 3417
                                                00000000G   00  16  0006F      JSB     FMG$MATCH_NAME                          ;
                                            5D              50  E9  00075      BLBC    R0, 3$                                  ;
                                        0748    C6              01  D0  00078 2$:  MOVL    #1, FOUND_MATCH                     ; 3421
                0084    8F          20          6E              00  2C  0007D  MOVC5   #0, (SP), #32, #132, DISBUF           ; 3427
                                                FF6C    C6          00084
                                        F8  A6      4F  A7  9B  00087          MOVZBW  LSTSTR2, FAODSC                         ; 3439
                                        FC  A6        50  A7  9E  0008C        MOVAB   LSTSTR2+1, FAODSC+4                     ;
                                                011C    C6  9F  00091          PUSHAB  RECBUF+148                              ;
                                                00FC    C6  9F  00095          PUSHAB  RECBUF+116                              ;
                                            7E  028C    C6  9A  00099          MOVZBL  RECBUF+516, -(SP)                       ;
                                                00AC    C6  DD  0009E          PUSHL   RECBUF+36                               ;
                                                0224    C6  9F  000A2          PUSHAB  RECBUF+412                              ;
                                    00000000V   00          02  FB  000A6      CALLS   #2, CLASSIFY_PRIV                       ;
                                                            50  DD  000AD      PUSHL   R0                                      ;
                                                00BC    C6  9F  000AF          PUSHAB  RECBUF+52                               ;
                                                            08  DD  000B3      PUSHL   #8                                      ;
                                                00AC    C6  DD  000B5          PUSHL   RECBUF+36                               ;
                                                008C    C6  9F  000B9          PUSHAB  RECBUF+4                                ;
                                                            0C  DD  000BD      PUSHL   #12                                     ;
                                                00DC    C6  9F  000BF          PUSHAB  RECBUF+84                               ;
                                                  F0    A6  9F  000C3          PUSHAB  DISDSC                                  ;
                                        7E              66  22  C1  000C6      ADDL3   #34, RABPTR, -(SP)                      ;
                                                  F8    A6  9F  000CA          PUSHAB  FAODSC                                  ;
                                            69              0D  FB  000CD      CALLS   #13, SYS$FAO                            ;
                                                            66  DD  000D0      PUSHL   RABPTR                                  ;
                                            68              01  FB  000D2      CALLS   #1, SYS$PUT                             ;
                                            50              01  D0  000D5 3$:  MOVL    #1, R0                                ; 3442
                                                            04  000D8          RET                                             ;

; Routine Size:  217 bytes,    Routine Base:  $CODE$ + 1531
```

```
;  3379          3443  1 %sbttl 'classify_priv - classifies contents of priv vector'
;  3380          3444  1 routine classify_priv (prvadr, uic) =
;  3381          3445  2 begin
;  3382          3446  2
;  3383          3447  2 !++
;  3384          3448  2 !
;  3385          3449  2 ! FUNCTIONAL DESCRIPTION:
;  3386          3450  2 !
;  3387          3451  2 !       Classifies privilege bits and reports the highest class available
;  3388          3452  2 !       to the owner of the supplied vector.
;  3389          3453  2 !
;  3390          3454  2 ! INPUTS:
;  3391          3455  2 !
;  3392          3456  2 !       PRVADR - Address of the privilege vector
;  3393          3457  2 !
;  3394          3458  2 ! IMPLICIT INPUTS:
;  3395          3459  2 !
;  3396          3460  2 !       none
;  3397          3461  2 !
;  3398          3462  2 ! OUTPUTS:
;  3399          3463  2 !
;  3400          3464  2 !       none
;  3401          3465  2 !
;  3402          3466  2 ! IMPLICIT OUTPUTS:
;  3403          3467  2 !
;  3404          3468  2 !       none
;  3405          3469  2 !
;  3406          3470  2 ! ROUTINE VALUE:
;  3407          3471  2 !
;  3408          3472  2 !       none
;  3409          3473  2 !
;  3410          3474  2 ! SIDE EFFECTS:
;  3411          3475  2 !
;  3412          3476  2 !       none
;  3413          3477  2 !--
;  3414          3478  2
;  3415          3479  2 map
;  3416          3480  2         prvadr  : ref block[8,byte];
;  3417          3481  2
;  3418          3482  2 bind
;  3419          3483  2         lstprva = cstring ('All'),
;  3420          3484  2         lstprvb = cstring ('Files'),
;  3421          3485  2         lstprvc = cstring ('System'),
;  3422          3486  2         lstprvd = cstring ('Devour'),
;  3423          3487  2         lstprve = cstring ('Group'),
;  3424          3488  2         lstprvf = cstring ('Normal'),
;  3425          3489  2         lstprvg = cstring ('None');
;  3426          3490  2
;  3427          3491  2 if .prvadr[prv$v_cmkrnl]
;  3428          3492  2 or .prvadr[prv$v_cmexec]
;  3429          3493  2 or .prvadr[prv$v_sysnam]
;  3430          3494  2 or .prvadr[prv$v_detach]
;  3431          3495  2 or .prvadr[prv$v_log_io]
;  3432          3496  2 or .prvadr[prv$v_setprv]
;  3433          3497  2 or .prvadr[prv$v_phy_io]
;  3434          3498  2 or .prvadr[prv$v_pfnmap]
;  3435          3499  2 or .prvadr[prv$v_sysprv]
```

```
;  3436          3500  2 or .prvadr[prv$v_readall]
;  3437          3501  2 or .prvadr[prv$v_bypass]
;  3438          3502  2 or (.uic<16, 16> lequ .EXE$GL_SYSUIC)
;  3439          3503  2 then return lstprva;                    ! Universal Privilege
;  3440          3504  2
;  3441          3505  2 if .prvadr[prv$v_diagnose]
;  3442          3506  2 or .prvadr[prv$v_volpro]
;  3443          3507  2 or .prvadr[prv$v_upgrade]
;  3444          3508  2 or .prvadr[prv$v_downgrade]
;  3445          3509  2 or .prvadr[prv$v_security]
;  3446          3510  2 or .prvadr[prv$v_sysgbl]
;  3447          3511  2 then return lstprvb;                    ! Potentially Comprimises File Security
;  3448          3512  2
;  3449          3513  2 if .prvadr[prv$v_pswapm]
;  3450          3514  2 or .prvadr[prv$v_setpri]
;  3451          3515  2 or .prvadr[prv$v_world]
;  3452          3516  2 or .prvadr[prv$v_oper]
;  3453          3517  2 then return lstprvc;                    ! Can Interfere with System Operation
;  3454          3518  2
;  3455          3519  2 if .prvadr[prv$v_grpnam]
;  3456          3520  2 or .prvadr[prv$v_allspool]
;  3457          3521  2 or .prvadr[prv$v_noacnt]
;  3458          3522  2 or .prvadr[prv$v_prmceb]
;  3459          3523  2 or .prvadr[prv$v_prmmbx]
;  3460          3524  2 or .prvadr[prv$v_exquota]
;  3461          3525  2 or .prvadr[prv$v_bugchk]
;  3462          3526  2 or .prvadr[prv$v_prmgbl]
;  3463          3527  2 or .prvadr[prv$v_shmem]
;  3464          3528  2 then return lstprvd;                    ! Can Devour System Resources
;  3465          3529  2
;  3466          3530  2 if .prvadr[prv$v_group]
;  3467          3531  2 or .prvadr[prv$v_grpprv]
;  3468          3532  2 then return lstprve;                    ! Can Interfere with Group Members
;  3469          3533  2
;  3470          3534  2 if .prvadr[prv$v_tmpmbx]
;  3471          3535  2 or .prvadr[prv$v_netmbx]
;  3472          3536  2 or .prvadr[prv$v_mount]
;  3473          3537  2 then return lstprvf;                    ! Normal Privileges
;  3474          3538  2
;  3475          3539  2 lstprvg                                 ! Not Privileged
;  3476          3540  1 end;

                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                 03 0031B P.ACU:  .BYTE   3
                              6C 6C 41 0031C      .ASCII  \All\                ;
                                 05 0031F P.ACV:  .BYTE   5
                           73 65 6C 69 46 00320      .ASCII  \Files\          ;
                                 06 00325 P.ACW:  .BYTE   6
                        6D 65 74 73 79 53 00326      .ASCII  \System\         ;
                                 06 0032C P.ACX:  .BYTE   6
                     72 75 6F 76 65 44 0032D      .ASCII  \Devour\            ;
                                 05 00333 P.ACY:  .BYTE   5
                        70 75 6F 72 47 00334      .ASCII  \Group\             ;
                                 06 00339 P.ACZ:  .BYTE   6
```

```
                     6C  61  6D  72  6F  4E  0033A                .ASCII  \Normal\                                        ;
                                     04  00340 P.ADA:    .BYTE   4                                                       ;
                                 65  6E  6F  4E  00341           .ASCII  \None\                                          ;

                                                    LSTPRVA=            P.ACU
                                                    LSTPRVB=            P.ACV
                                                    LSTPRVC=            P.ACW
                                                    LSTPRVD=            P.ACX
                                                    LSTPRVE=            P.ACY
                                                    LSTPRVF=            P.ACZ
                                                    LSTPRVG=            P.ADA

                                                                .PSECT  $CODE$,NOWRT,2

                                             0004 00000 CLASSIFY_PRIV:
                                                                .WORD   Save R2                                         ; 3444
                         52 00000000'    00  9E 00002           MOVAB   LSTPRVA, R2                                     ;
                         50       04     AC  D0 00009           MOVL    PRVADR, R0                                      ; 3491
                         35              60  E8 0000D           BLBS    (R0), 1$                                        ;
                 31      60              01  E0 00010           BBS     #1, (R0), 1$                                    ; 3492
                 2D      60              02  E0 00014           BBS     #2, (R0), 1$                                    ; 3493
                 29      60              05  E0 00018           BBS     #5, (R0), 1$                                    ; 3494
                                         60  95 0001C           TSTB    (R0)                                            ; 3495
                                         25  19 0001E           BLSS    1$                                              ;
                 21      60              0E  E0 00020           BBS     #14, (R0), 1$                                   ; 3496
                 1D      60              16  E0 00024           BBS     #22, (R0), 1$                                   ; 3497
                 19      60              1A  E0 00028           BBS     #26, (R0), 1$                                   ; 3498
                 15      60              1C  E0 0002C           BBS     #28, (R0), 1$                                   ; 3499
                 10      04      A0      03  E0 00030           BBS     #3, 4(R0), 1$                                   ; 3500
                 0C      60              1D  E0 00035           BBS     #29, (R0), 1$                                   ; 3501
00000000G  00    0A      AC              10  00 ED 00039        CMPZV   #0, #16, UIC+2, EXE$GL_SYSUIC                   ; 3502
                                         05  1A 00043           BGTRU   2$                                              ;
                         51              62  9E 00045 1$:      MOVAB   LSTPRVA, R1                                     ; 3503
                                         6D  11 00048           BRB     10$                                             ;
                 16      60              06  E0 0004A 2$:      BBS     #6, (R0), 3$                                    ; 3505
                 12      60              15  E0 0004E           BBS     #21, (R0), 3$                                   ; 3506
                         0E      04      A0  E8 00052           BLBS    4(R0), 3$                                       ; 3507
                 09      04      A0      01  E0 00056           BBS     #1, 4(R0), 3$                                   ; 3508
                 04      04      A0      06  E0 0005B           BBS     #6, 4(R0), 3$                                   ; 3509
                 06      60              19  E1 00060           BBC     #25, (R0), 4$                                   ; 3510
                         51      04      A2  9E 00064 3$:      MOVAB   LSTPRVB, R1                                     ; 3511
                                         4D  11 00068           BRB     10$                                             ;
                 0C      60              0C  E0 0006A 4$:      BBS     #12, (R0), 5$                                   ; 3513
                 08      60              0D  E0 0006E           BBS     #13, (R0), 5$                                   ; 3514
                         04      02      A0  E8 00072           BLBS    2(R0), 5$                                       ; 3515
                 06      60              12  E1 00076           BBC     #18, (R0), 6$                                   ; 3516
                         51      0A      A2  9E 0007A 5$:      MOVAB   LSTPRVC, R1                                     ; 3517
                                         37  11 0007E           BRB     10$                                             ;
                 20      60              03  E0 00080 6$:      BBS     #3, (R0), 7$                                    ; 3519
                 1C      60              04  E0 00084           BBS     #4, (R0), 7$                                    ; 3520
                 18      60              09  E0 00088           BBS     #9, (R0), 7$                                    ; 3521
                 14      60              0A  E0 0008C           BBS     #10, (R0), 7$                                   ; 3522
                 10      60              0B  E0 00090           BBS     #11, (R0), 7$                                   ; 3523
                 0C      60              13  E0 00094           BBS     #19, (R0), 7$                                   ; 3524
                 08      60              17  E0 00098           BBS     #23, (R0), 7$                                   ; 3525
                         04      03      A0  E8 0009C           BLBS    3(R0), 7$                                       ; 3526
```

```
                06              60          1B  E1  000A0          BBC     #27, (R0), 8$                   ; 3527
                                51      11  A2  9E  000A4  7$:     MOVAB   LSTPRVD, R1                     ; 3528
                                            0D  11  000A8          BRB     10$
                            05  01      A0  E8  000AA  8$:      BLBS    1(R0), 9$                       ; 3530
                08      04  A0          02  E1  000AE          BBC     #2, 4(R0), 11$                  ; 3531
                                51      18  A2  9E  000B3  9$:      MOVAB   LSTPRVE, R1                     ; 3532
                                50          51  D0  000B7  10$:     MOVL    R1, R0
                                            04  000BA          RET
                                        60  B5  000BB  11$:     TSTW    (R0)                            ; 3534
                                        08  19  000BD          BLSS    12$
                04              60          14  E0  000BF          BBS     #20, (R0), 12$                  ; 3535
                05              60          11  E1  000C3          BBC     #17, (R0), 13$                  ; 3536
                                50      1E  A2  9E  000C7  12$:     MOVAB   LSTPRVF, R0                     ; 3537
                                            04  000CB          RET
                                50      25  A2  9E  000CC  13$:     MOVAB   LSTPRVG, R0                     ; 3445
                                            04  000D0          RET                                     ; 3540

; Routine Size:  209 bytes,    Routine Base:  $CODE$ + 160A
```

```
;  3478     3541  1 %sbttl 'display_full - writes the full user display'
;  3479     3542  1 routine display_full =
;  3480     3543  2 begin
;  3481     3544  2
;  3482     3545  2 !++
;  3483     3546  2 !
;  3484     3547  2 ! FUNCTIONAL DESCRIPTION:
;  3485     3548  2 !
;  3486     3549  2 !       Display the fields of a UAF record.
;  3487     3550  2 !
;  3488     3551  2 ! INPUTS:
;  3489     3552  2 !
;  3490     3553  2 !       RABPTR - RMS data structure for the file
;  3491     3554  2 !
;  3492     3555  2 ! IMPLICIT INPUTS:
;  3493     3556  2 !
;  3494     3557  2 !       none
;  3495     3558  2 !
;  3496     3559  2 ! OUTPUTS:
;  3497     3560  2 !
;  3498     3561  2 !       none
;  3499     3562  2 !
;  3500     3563  2 ! IMPLICIT OUTPUTS:
;  3501     3564  2 !
;  3502     3565  2 !       none
;  3503     3566  2 !
;  3504     3567  2 ! ROUTINE VALUE:
;  3505     3568  2 !
;  3506     3569  2 !       none
;  3507     3570  2 !
;  3508     3571  2 ! SIDE EFFECTS:
;  3509     3572  2 !
;  3510     3573  2 !       none
;  3511     3574  2 !--
;  3512     3575  2
;  3513     3576  2 !
;  3514     3577  2 ! Display strings.
;  3515     3578  2 !
;  3516     3579  2
;  3517     3580  2 local
;  3518     3581  2     status      : long ;
;  3519     3582  2
;  3520     3583  2 bind
;  3521     3584  2     username    = cstring ('Username: !32AF Owner:  !AC'),
;  3522     3585  2     account     = cstring ('Account:  !32AF UIC:    !%U (!%I)'),
;  3523     3586  2     cli_table   = cstring ('CLI:      !32AC Tables: !AC'),
;  3524     3587  2     default     = cstring ('Default:  !AC!AC'),
;  3525     3588  2     lgicmd      = cstring ('LGICMD:   !AC'),
;  3526     3589  2     flags       = cstring ('Login Flags: !AD')
;  3527     3590  2     flag_pad    = cstring (%char (cr), %char (lf), '                '),
;  3528     3591  2     primdays    = cstring ('Primary days:  !7(AC)'),
;  3529     3592  2     secdays     = cstring ('Secondary days:!7(AC)'),
;  3530     3593  2     norestrict  = cstring ('No access restrictions'),
;  3531  P  3594  2     accesshdr1  = cstring (
;  3532     3595  2 'Primary   000000000011111111112222  Secondary 000000000011111111112222'),
;  3533  P  3596  2     accesshdr2  = cstring (
;  3534     3597  2 'Day Hours 012345678901234567890123  Day Hours 012345678901234567890123'),
```

```
;  3535     3598  2      netaccess   = cstring ('Network:   !AD            !AD'),
;  3536     3599  2      bataccess   = cstring ('Batch:     !AD            !AD'),
;  3537     3600  2      locaccess   = cstring ('Local:     !AD            !AD'),
;  3538     3601  2      diaaccess   = cstring ('Dialup:    !AD            !AD'),
;  3539     3602  2      remaccess   = cstring ('Remote:    !AD            !AD'),
;  3540     3603  2      expiration  = cstring ('Expiration: !AD    Pwdminimum: !2UL   Login Fails: !5UL'),
;  3541     3604  2      pwddata     = cstring ('Pwdlifetime:       !AD    Pwdchange:  !AD !AD'),
;  3542     3605  2      lastlogin   = cstring ('Last Login: !AD (interactive), !AD (non-interactive)'),
;  3543     3606  2      quota1      = cstring ('Maxjobs:     !5UL  Fillm:     !5UL  Bytlm:    !9UL'),
;  3544     3607  2      quota2      = cstring ('Maxacctjobs: !5UL  Shrfillm:  !5UL  Pbytlm:   !9UL'),
;  3545     3608  2      quota3      = cstring ('Maxdetach:   !5UL  BIOlm:     !5UL  JTquota:  !9UL'),
;  3546     3609  2      quota4      = cstring ('Prclm:       !5UL  DIOlm:     !5UL  WSdef:    !9UL'),
;  3547     3610  2      quota5      = cstring ('Prio:        !5UL  ASTlm:     !5UL  WSquo:    !9UL'),
;  3548     3611  2      quota6      = cstring ('Queprio:     !5UL  TQElm:     !5UL  WSextent: !9UL'),
;  3549     3612  2      quota7      = cstring ('CPU: !13AD  Enqlm:     !5UL  Pgflquo:  !9UL'),
;  3550     3613  2      privs       = cstring ('Authorized Privileges: '),
;  3551     3614  2      defprivs    = cstring ('Default Privileges: '),
;  3552     3615  2
;  3553     3616  2      nullstr     = cstring (''),
;  3554     3617  2
;  3555     3618  2      mon         = cstring (' Mon'),
;  3556     3619  2      tue         = cstring (' Tue'),
;  3557     3620  2      wed         = cstring (' Wed'),
;  3558     3621  2      thu         = cstring (' Thu'),
;  3559     3622  2      fri         = cstring (' Fri'),
;  3560     3623  2      sat         = cstring (' Sat'),
;  3561     3624  2      sun         = cstring (' Sun'),
;  3562     3625  2      noday       = cstring ('    '),
;  3563     3626  2
;  3564     3627  2      cputime     = recbuf[uaf$l_cputim]; ! CPU limit in hundredths of a second
;  3565     3628  2
;  3566     3629  2  own
;  3567     3630  2      flags_vector : vector [32] preset (
;  3568     3631  2      [$bitposition (uaf$v_audit)]        = cstring (' Audit'),
;  3569     3632  2      [$bitposition (uaf$v_captive)]      = cstring (' Captive'),
;  3570     3633  2      [$bitposition (uaf$v_defcli)]       = cstring (' Defcli'),
;  3571     3634  2      [$bitposition (uaf$v_disctly)]      = cstring (' Disctly'),
;  3572     3635  2      [$bitposition (uaf$v_nomail)]       = cstring (' Dismail'),
;  3573     3636  2      [$bitposition (uaf$v_dismail)]      = cstring (' Disnewmail'),
;  3574     3637  2      [$bitposition (uaf$v_disreconnect)] = cstring (' Disreconnect'),
;  3575     3638  2      [$bitposition (uaf$v_disreport)]    = cstring (' Disreport'),
;  3576     3639  2      [$bitposition (uaf$v_disacnt)]      = cstring (' Disuser'),
;  3577     3640  2      [$bitposition (uaf$v_diswelcom)]    = cstring (' Diswelcome'),
;  3578     3641  2      [$bitposition (uaf$v_genpwd)]       = cstring (' Genpwd'),
;  3579     3642  2      [$bitposition (uaf$v_lockpwd)]      = cstring (' Lockpwd'),
;  3580     3643  2      [$bitposition (uaf$v_pwd_expired)]  = cstring (' Pwd_expired'),
;  3581     3644  3      [$bitposition (uaf$v_pwd2_expired)] = cstring (' Pwd2_expired')
;  3582     3645  2      );
;  3583     3646  2
;  3584     3647  2  local
;  3585     3648  2          count,                                  ! count for string being built
;  3586     3649  2          lcount,                                 ! count of chars on current line
;  3587     3650  2          string          : vector [160, byte],   ! buffer to build display string
;  3588     3651  2          flag_string     : ref vector [,byte],   ! pointer to flag string
;  3589     3652  2          delta_time      : vector [long, 2],     ! Scratch area for system delta time
;  3590     3653  2          PTR,                                    ! Pointer into UAF$Q_PWD_DATE quadword
;  3591     3654  2          time1           : vector [17, byte],    ! buffer for time string
```

```
;  3592          3655  2          time2           : vector [17, byte],    ! buffer for time string
;  3593          3656  2          time3           : vector [17, byte];    ! buffer for time string
;  3594          3657  2
;  3595          3658  2  builtin
;  3596          3659  2          emul;
;  3597          3660  2
;  3598          3661  2
;  3599          3662  2
;  3600          3663  2  if .str_wild and not fmg$match_name (namelen (uaf$s_username,recbuf[uaf$t_username]),
;  3601          3664  2                                      recbuf[uaf$t_username],
;  3602          3665  2                                      .match_token[en,  match_token)
;  3603          3666  2  then return true;
;  3604          3667  2
;  3605          3668  2  found_match = true;
;  3606          3669  2
;  3607          3670  2  output_null;
;  3608          3671  2
;  3609        P 3672  2  faomac (username,
;  3610        P 3673  2      uaf$s_username, recbuf[uaf$t_username],
;  3611          3674  2      recbuf[uaf$t_owner]);
;  3612          3675  2
;  3613        P 3676  2  faomac (account,
;  3614        P 3677  2      uaf$s_account, recbuf[uaf$t_account],
;  3615        P 3678  2      .recbuf[uaf$l_uic],
;  3616          3679  2      .recbuf[uaf$l_uic]);
;  3617          3680  2
;  3618        P 3681  2  faomac (cli_table,
;  3619        P 3682  2      recbuf[uaf$t_defcli],
;  3620          3683  2      recbuf[uaf$t_clitables]);
;  3621          3684  2
;  3622        P 3685  2  faomac (default,
;  3623        P 3686  2      recbuf[uaf$t_defdev],
;  3624          3687  2      recbuf[uaf$t_defdir]);
;  3625          3688  2
;  3626        P 3689  2  faomac (lgicmd,
;  3627          3690  2      recbuf[uaf$t_lgicmd]);
;  3628          3691  2
;  3629          3692  2  count = 0;
;  3630          3693  2  lcount = .flags<0,8>;
;  3631          3694  2  incr j from 0 to 31
;  3632          3695  2  do
;  3633          3696  3      begin
;  3634          3697  3      if .bitvector [recbuf[uaf$l_flags], .j]
;  3635          3698  3      and (flag_string = .flags_vector [.j]) neq 0
;  3636          3699  3      then
;  3637          3700  4          begin
;  3638          3701  4          if .lcount + .flag_string[0] gtru 80
;  3639          3702  4          then
;  3640          3703  5              begin
;  3641          3704  5              ch$move (.flag_pad<0,8>, flag_pad+1, string[.count]);
;  3642          3705  5              count = .count + .flag_pad<0,8>;
;  3643          3706  5              lcount = .flag_pad<0,8> - 2;
;  3644          3707  4              end;
;  3645          3708  4          ch$move (.flag_string[0], flag_string[1], string[.count]);
;  3646          3709  4          count = .count + .flag_string[0];
;  3647          3710  4          lcount = .lcount + .flag_string[0];
;  3648          3711  3          end;
```

```
;  3649          3712  2     end;
;  3650          3713
;  3651        P 3714  2  faomac (flags,
;  3652          3715  2      .count, string);
;  3653          3716
;  3654        P 3717  2  faomac (primdays,
;  3655        P 3718  2      if .recbuf[uaf$v_monday]      then noday   else mon,
;  3656        P 3719  2      if .recbuf[uaf$v_tuesday]     then noday   else tue,
;  3657        P 3720  2      if .recbuf[uaf$v_wednesday]   then noday   else wed,
;  3658        P 3721  2      if .recbuf[uaf$v_thursday]    then noday   else thu,
;  3659        P 3722  2      if .recbuf[uaf$v_friday]      then noday   else fri,
;  3660        P 3723  2      if .recbuf[uaf$v_saturday]    then noday   else sat,
;  3661          3724  2      if .recbuf[uaf$v_sunday]      then noday   else sun);
;  3662          3725
;  3663        P 3726  2  faomac (secdays,
;  3664        P 3727  2      if .recbuf[uaf$v_monday]      then mon     else noday,
;  3665        P 3728  2      if .recbuf[uaf$v_tuesday]     then tue     else noday,
;  3666        P 3729  2      if .recbuf[uaf$v_wednesday]   then wed     else noday,
;  3667        P 3730  2      if .recbuf[uaf$v_thursday]    then thu     else noday,
;  3668        P 3731  2      if .recbuf[uaf$v_friday]      then fri     else noday,
;  3669        P 3732  2      if .recbuf[uaf$v_saturday]    then sat     else noday,
;  3670          3733  2      if .recbuf[uaf$v_sunday]      then sun     else noday);
;  3671          3734
;  3672          3735  2  if ch$fail (ch$find_not_ch (10*uaf$s_network_access_p, recbuf[uaf$b_network_access_p], 0))
;  3673          3736  2  then
;  3674          3737  2      begin
;  3675          3738  2      faomac (norestrict);
;  3676          3739  2      end
;  3677          3740  2  else
;  3678          3741  2      begin
;  3679          3742  2      faomac (accesshdr1);
;  3680          3743  2      faomac (accesshdr2);
;  3681          3744  2
;  3682          3745  2      display_hours (netaccess, recbuf[uaf$b_network_access_p]);
;  3683          3746  2      display_hours (bataccess, recbuf[uaf$b_batch_access_p]);
;  3684          3747  2      display_hours (locaccess, recbuf[uaf$b_local_access_p]);
;  3685          3748  2      display_hours (diaaccess, recbuf[uaf$b_dialup_access_p]);
;  3686          3749  2      display_hours (remaccess, recbuf[uaf$b_remote_access_p]);
;  3687          3750  2      end;
;  3688          3751  2
;  3689          3752  2  convert_time (recbuf[uaf$q_expiration], 17, time1);
;  3690        P 3753  2  faomac (expiration,
;  3691        P 3754  2      17, time1,
;  3692        P 3755  2      .recbuf[uaf$b_pwd_length],
;  3693          3756  2      .recbuf[uaf$w_logfails]);
;  3694          3757  2
;  3695          3758  2  convert_time (recbuf[uaf$q_pwd_lifetime], 10, time1);
;  3696          3759  2  PTR = RECBUF[UAF$Q_PWD_DATE];   ! Because quadwords have "no" width
;  3697          3760  2  if (..PTR eql -1) and (.(.PTR+%upval) eql -1) then
;  3698          3761  2     ch$move(17, uplit(%ascii'    (pre-expired)'), TIME2)
;  3699          3762  2  else
;  3700          3763  2     CONVERT_TIME(.PTR, 17, TIME2);
;  3701          3764  2  convert_time (recbuf[uaf$q_pwd2_date], 17, time3);
;  3702        P 3765  2  faomac (pwddata,
;  3703        P 3766  2      10, time1,
;  3704        P 3767  2      17, time2,
;  3705        P 3768  2      (if (. (recbuf[uaf$q_pwd2_date]+0) or . (recbuf[uaf$q_pwd2_date]+4)) eql 0
```

```
;  3706     P 3769  2          then 0
;  3707       3770  2          else 17), time3);
;  3708       3771  2
;  3709       3772  2      convert_time (recbuf[uaf$q_lastlogin_i], 17, time1);
;  3710       3773  2      convert_time (recbuf[uaf$q_lastlogin_n], 17, time2);
;  3711     P 3774  2      faomac (lastlogin,
;  3712     P 3775  2          17, time1,
;  3713       3776  2          17, time2);
;  3714       3777  2
;  3715       3778  2      emul (%ref (-200000), %ref (.cputime<1,31>),
;  3716       3779  2            %ref (if .cputime<0,1> then -100000 else 0), delta_time);
;  3717       3780  2      convert_time (delta_time, 13, time1);
;  3718       3781  2
;  3719     P 3782  2      faomac (quota1,
;  3720     P 3783  2          .recbuf[uaf$w_maxjobs],
;  3721     P 3784  2          .recbuf[uaf$w_fillm],
;  3722       3785  2          .recbuf[uaf$l_bytlm]);
;  3723       3786  2
;  3724     P 3787  2      faomac (quota2,
;  3725     P 3788  2          .recbuf[uaf$w_maxacctjobs],
;  3726     P 3789  2          .recbuf[uaf$w_shrfillm],
;  3727       3790  2          .recbuf[uaf$l_pbytlm]);
;  3728       3791  2
;  3729     P 3792  2      faomac (quota3,
;  3730     P 3793  2          .recbuf[uaf$w_maxdetach],
;  3731     P 3794  2          .recbuf[uaf$w_biolm],
;  3732       3795  2          .recbuf[uaf$l_jtquota]);
;  3733       3796  2
;  3734     P 3797  2      faomac (quota4,
;  3735     P 3798  2          .recbuf[uaf$w_prccnt],
;  3736     P 3799  2          .recbuf[uaf$w_diolm],
;  3737       3800  2          .recbuf[uaf$l_dfwscnt]);
;  3738       3801  2
;  3739     P 3802  2      faomac (quota5,
;  3740     P 3803  2          .recbuf[uaf$b_pri],
;  3741     P 3804  2          .recbuf[uaf$w_astlm],
;  3742       3805  2          .recbuf[uaf$l_wsquota]);
;  3743       3806  2
;  3744     P 3807  2      faomac (quota6,
;  3745     P 3808  2          .recbuf[uaf$b_quepri],
;  3746     P 3809  2          .recbuf[uaf$w_tqcnt],
;  3747       3810  2          .recbuf[uaf$l_wsextent]);
;  3748       3811  2
;  3749     P 3812  2      faomac (quota7,
;  3750     P 3813  2          13, time1,
;  3751     P 3814  2          .recbuf[uaf$w_englm],
;  3752       3815  2          .recbuf[uaf$l_pgflquota]);
;  3753       3816  2
;  3754       3817  2      faomac (privs);
;  3755       3818  2      print_priv (recbuf[uaf$q_priv]);
;  3756       3819  2
;  3757       3820  2      faomac (defprivs);
;  3758       3821  2      print_priv (recbuf[uaf$q_def_priv]);
;  3759       3822  2
;  3760       3823  2      !
;  3761       3824  2      ! Build a holder from the UAF record and display the rights
;  3762       3825  2      ! granted to it.
```

```
;  3763          3826  2 !
;  3764          3827  2 if .rdb_exists
;  3765          3828  2 then
;  3766          3829  2     begin
;  3767          3830  2     uaf$build_holder ();
;  3768          3831  2     rdb_header_flag = true ;
;  3769          3832  2     status = uaf$write_rights ( holder ) ;
;  3770          3833  2     end ;
;  3771          3834  2
;  3772          3835  2 return .status ;
;  3773          3836  1 end;


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                    1B 00345 P.ADB:  .BYTE   27
46 41 32 33 21 20 3A 65 6D 61 6E 72 65 73           55 00346         .ASCII  \Username: !32AF Owner:  !AC\
                43 41 21 20 20 3A 72 65 6E 77 4F    20 00355
                                                    21 00361 P.ADC:  .BYTE   33
46 41 32 33 21 20 20 3A 74 6E 75 6F 63 63           41 00362         .ASCII  \Account:  !32AF UIC:    !%U (!%I)\
21 28 20 55 25 21 20 20 20 20 3A 43 49 55           20 00371
                                  29 49             25 00380
                                                    1B 00383 P.ADD:  .BYTE   27
43 41 32 33 21 20 20 20 20 20 20 3A 49 4C           43 00384         .ASCII  \CLI:       !32AC Tables: !AC\
                43 41 21 20 3A 73 65 6C 62 61 54    20 00393
                                                    10 0039F P.ADE:  .BYTE   16
41 21 43 41 21 20 20 3A 74 6C 75 61 66 65           44 003A0         .ASCII  \Default:  !AC!AC\
                                                    43 003AF
                                                    0D 003B0 P.ADF:  .BYTE   13
      43 41 21 20 20 20 3A 44 4D 43 49 47           4C 003B1         .ASCII  \LGICMD:   !AC\
                                                    10 003BE P.ADG:  .BYTE   16
41 21 20 3A 73 67 61 6C 46 20 6E 69 67 6F           4C 003BF         .ASCII  \Login Flags: !AD\
                                                    44 003CE
                                                    0F 003CF P.ADH:  .BYTE   15
20 20 20 20 20 20 20 20 20 20 20 20 20 0A           0D 003D0         .ASCII  <13><10>\               \
                                                    15 003DF P.ADI:  .BYTE   21
20 20 3A 73 79 61 64 20 79 72 61 6D 69 72           50 003E0         .ASCII  \Primary days:  !7(AC)\
                              29 43 41 28 37        21 003EF
                                                    15 003F5 P.ADJ:  .BYTE   21
3A 73 79 61 64 20 79 72 61 64 6E 6F 63 65           53 003F6         .ASCII  \Secondary days:!7(AC)\
                              29 43 41 28 37        21 00405
                                                    16 00408 P.ADK:  .BYTE   22
72 74 73 65 72 20 73 73 65 63 63 61 20 6F           4E 0040C         .ASCII  \No access restrictions\
                              73 6E 6F 69 74 63     69 0041B
                                                    46 00422 P.ADL:  .BYTE   70
30 30 30 30 30 20 20 20 79 72 61 6D 69 72           50 00423         .ASCII  \Primary   00000000001111111111222  Seco\
31 31 31 31 31 31 31 31 31 30 30 30 30 30           30 00432
                  6F 63 65 53 20 20 32 32 32        32 00441
30 30 30 30 30 30 30 30 30 20 79 72 61 64           6E 0044B         .ASCII  \ndary 00000000001111111111222\
32 32 32 32 31 31 31 31 31 31 31 31 31 31           30 0045A
                                                    46 00469 P.ADM:  .BYTE   70
34 33 32 31 30 20 73 72 75 6F 48 20 79 61           44 0046A         .ASCII  \Day Hours 012345678901234567890123  Day \
39 38 37 36 35 34 33 32 31 30 39 38 37 36           35 00479
                  20 79 61 44 20 20 33 32 31        30 00488
38 37 36 35 34 33 32 31 30 20 73 72 75 6F           48 00492         .ASCII  \Hours 012345678901234567890123\
33 32 31 30 39 38 37 36 35 34 33 32 31 30           39 004A1
```

```
                                                        1C  004B0 P.ADN:  .BYTE   28
20  20  44  41  21  20  20  3A  6B  72  6F  77  74  65  4E  004B1         .ASCII  \Network:  !AD              !AD\
        44  41  21  20  20  20  20  20  20  20  20  20  20  004C0
                                                        1C  004CD P.ADO:  .BYTE   28
20  20  44  41  21  20  20  20  20  3A  68  63  74  61  42  004CE         .ASCII  \Batch:    !AD              !AD\
        44  41  21  20  20  20  20  20  20  20  20  20  20  004DD
                                                        1C  004EA P.ADP:  .BYTE   28
20  20  44  41  21  20  20  20  20  3A  6C  61  63  6F  4C  004EB         .ASCII  \Local:    !AD              !AD\
        44  41  21  20  20  20  20  20  20  20  20  20  20  004FA
                                                        1C  00507 P.ADQ:  .BYTE   28
20  20  44  41  21  20  20  20  3A  70  75  6C  61  69  44  00508         .ASCII  \Dialup:   !AD              !AD\
        44  41  21  20  20  20  20  20  20  20  20  20  20  00517
                                                        1C  00524 P.ADR:  .BYTE   28
20  20  44  41  21  20  20  20  3A  65  74  6F  6D  65  52  00525         .ASCII  \Remote:   !AD              !AD\
        44  41  21  20  20  20  20  20  20  20  20  20  20  00534
                                                        37  00541 P.ADS:  .BYTE   55
44  41  21  20  3A  6E  6F  69  74  61  72  69  70  78  45  00542         .ASCII  \Expiration: !AD   Pwdminimum: !2UL   Lo\
3A  6D  75  6D  69  6E  69  6D  64  77  50  20  20  20  20  00551
                    6F  4C  20  20  20  4C  55  32  21  20  00560
4C  55  35  21  20  3A  73  6C  69  61  46  20  6E  69  67  0056A         .ASCII  \gin Fails: !5UL\
                                                        2D  00579 P.ADT:  .BYTE   45
20  20  20  3A  65  6D  69  74  65  66  69  6C  64  77  50  0057A         .ASCII  \Pwdlifetime:        !AD   Pwdchange:  !A\
63  64  77  50  20  20  20  20  44  41  21  20  20  20  20  00589
                    41  21  20  20  3A  65  67  6E  61  68  00598
                                        44  41  21  20  44  005A2         .ASCII  \D !AD\
                                                        34  005A7 P.ADU:  .BYTE   52
44  41  21  20  3A  6E  69  67  6F  4C  20  74  73  61  4C  005A8         .ASCII  \Last Login: !AD (interactive), !AD (non-\
2C  29  65  76  69  74  63  61  72  65  74  6E  69  28  20  005B7
                    2D  6E  6F  6E  28  20  44  41  21  20  005C6
            29  65  76  69  74  63  61  72  65  74  6E  69  005D0         .ASCII  \interactive)\
                                                        32  005DC P.ADV:  .BYTE   50
35  21  20  20  20  20  20  3A  73  62  6F  6A  78  61  4D  005DD         .ASCII  \Maxjobs:      !5UL  Fillm:      !5UL  Bytl\
20  20  20  20  20  3A  6D  6C  6C  69  46  20  20  4C  55  005EC
                    6C  74  79  42  20  20  4C  55  35  21  005FB
                    4C  55  39  21  20  20  20  20  3A  6D  00605         .ASCII  \m:     !9UL\
                                                        32  0060F P.ADW:  .BYTE   50
35  21  20  3A  73  62  6F  6A  74  63  63  61  78  61  4D  00610         .ASCII  \Maxacctjobs: !5UL  Shrfillm:  !5UL  Pbyt\
20  20  3A  6D  6C  6C  69  66  72  68  53  20  20  4C  55  0061F
                    74  79  62  50  20  20  4C  55  35  21  0062E
                    4C  55  39  21  20  20  20  3A  6D  6C  00638         .ASCII  \lm:   !9UL\
                                                        32  00642 P.ADX:  .BYTE   50
35  21  20  20  20  3A  68  63  61  74  65  64  78  61  4D  00643         .ASCII  \Maxdetach:   !5UL  BIOlm:      !5UL  JTqu\
20  20  20  20  20  3A  6D  6C  4F  49  42  20  20  4C  55  00652
                    75  71  54  4A  20  20  4C  55  35  21  00661
                    4C  55  39  21  20  20  3A  61  74  6F  0066B         .ASCII  \ota:  !9UL\
                                                        32  00675 P.ADY:  .BYTE   50
35  21  20  20  20  20  20  20  20  3A  6D  6C  63  72  50  00676         .ASCII  \Prclm:        !5UL  DIOlm:      !5UL  WSde\
20  20  20  20  20  3A  6D  6C  4F  49  44  20  20  4C  55  00685
                    65  64  53  57  20  20  4C  55  35  21  00694
                    4C  55  39  21  20  20  20  20  3A  66  0069E         .ASCII  \f:     !9UL\
                                                        32  006A8 P.ADZ:  .BYTE   50
35  21  20  20  20  20  20  20  20  20  3A  6F  69  72  50  006A9         .ASCII  \Prio:         !5UL  ASTlm:      !5UL  WSqu\
20  20  20  20  20  3A  6D  6C  54  53  41  20  20  4C  55  006B8
                    75  71  53  57  20  20  4C  55  35  21  006C7
                    4C  55  39  21  20  20  20  20  3A  6F  006D1         .ASCII  \o:     !9UL\
                                                        32  006DB P.AEA:  .BYTE   50
35  21  20  20  20  20  20  3A  6F  69  72  70  65  75  51  006DC         .ASCII  \Queprio:      !5UL  TQElm:      !5UL  WSex\
```

```
20 20 20 20 20 3A 6D 6C 45 51 54 20 20 4C 55 006EB
               78 65 53 57 20 20 4C 55 35 21 006FA
               4C 55 39 21 20 3A 74 6E 65 74 00704                 .ASCII  \tent: !9UL\
                                          2B 0070E P.AEB:          .BYTE   43
71 6E 45 20 20 44 41 33 31 21 20 3A 55 50 43 0070F                 .ASCII  \CPU: !13AD  Enqlm:     !5UL  Pgflquo:  !\
50 20 20 4C 55 35 21 20 20 20 20 20 3A 6D 6C 0071E
               21 20 20 3A 6F 75 71 6C 66 67 0072D
                                  4C 55 39 00737                 .ASCII  \9UL\
                                          17 0073A P.AEC:          .BYTE   23
76 69 72 50 20 64 65 7A 69 72 6F 68 74 75 41 0073B                 .ASCII  \Authorized Privileges: \
                     20 3A 73 65 67 65 6C 69 0074A
                                          14 00752 P.AED:          .BYTE   20
65 6C 69 76 69 72 50 20 74 6C 75 61 66 65 44 00753                 .ASCII  \Default Privileges: \
                           20 3A 73 65 67 00762
                                          00 00767 P.AEE:          .BYTE   0
                                             00768                 .BLKB   0
                                          04 00768 P.AEF:          .BYTE   4
                                 6E 6F 4D 20 00769                 .ASCII  \ Mon\
                                          04 0076D P.AEG:          .BYTE   4
                                 65 75 54 20 0076E                 .ASCII  \ Tue\
                                          04 00772 P.AEH:          .BYTE   4
                                 64 65 57 20 00773                 .ASCII  \ Wed\
                                          04 00777 P.AEI:          .BYTE   4
                                 75 68 54 20 00778                 .ASCII  \ Thu\
                                          04 0077C P.AEJ:          .BYTE   4
                                 69 72 46 20 0077D                 .ASCII  \ Fri\
                                          04 00781 P.AEK:          .BYTE   4
                                 74 61 53 20 00782                 .ASCII  \ Sat\
                                          04 00786 P.AEL:          .BYTE   4
                                 6E 75 53 20 00787                 .ASCII  \ Sun\
                                          04 0078B P.AEM:          .BYTE   4
                                 20 20 20 20 0078C                 .ASCII  \    \
                                          06 00790 P.AEN:          .BYTE   6
                              74 69 64 75 41 20 00791                 .ASCII  \ Audit\
                                          08 00797 P.AEO:          .BYTE   8
                        65 76 69 74 70 61 43 20 00798                 .ASCII  \ Captive\
                                          07 007A0 P.AEP:          .BYTE   7
                           69 6C 63 66 65 44 20 007A1                 .ASCII  \ Defcli\
                                          08 007A8 P.AEQ:          .BYTE   8
                        79 6C 74 63 73 69 44 20 007A9                 .ASCII  \ Disctly\
                                          08 007B1 P.AER:          .BYTE   8
                        6C 69 61 6D 73 69 44 20 007B2                 .ASCII  \ Dismail\
                                          0B 007BA P.AES:          .BYTE   11
                  6C 69 61 6D 77 65 6E 73 69 44 20 007BB                 .ASCII  \ Disnewmail\
                                          0D 007C6 P.AET:          .BYTE   13
74 63 65 6E 6E 6F 63 65 72 73 69 44 20 007C7                 .ASCII  \ Disreconnect\
                                          0A 007D4 P.AEU:          .BYTE   10
               74 72 6F 70 65 72 73 69 44 20 007D5                 .ASCII  \ Disreport\
                                          08 007DF P.AEV:          .BYTE   8
                        72 65 73 75 73 69 44 20 007E0                 .ASCII  \ Disuser\
                                          0B 007E8 P.AEW:          .BYTE   11
                  65 6D 6F 63 6C 65 77 73 69 44 20 007E9                 .ASCII  \ Diswelcome\
                                          07 007F4 P.AEX:          .BYTE   7
                           64 77 70 6E 65 47 20 007F5                 .ASCII  \ Genpwd\
                                          08 007FC P.AEY:          .BYTE   8
                        64 77 70 6B 63 6F 4C 20 007FD                 .ASCII  \ Lockpwd\
                                          0C 00805 P.AEZ:          .BYTE   12
```

```
                64  65  72  69  70  78  65  5F  64  77  50  20  00806                .ASCII  \ Pwd_expired\                          ;
                                                    0D  00812 P.AFA:    .BYTE   13                                               ;
            64  65  72  69  70  78  65  5F  32  64  77  50  20  00813                .ASCII  \ Pwd2_expired\                         ;
65  72  69 70  78  65  2D  65  72  70  28  20  20  20  20  00820 P.AFB:    .ASCII  \    (pre-expired)\<0><0><0>                    ;
                                        00  00  00  29  64  0082F                                                                ;

                                                                                .PSECT  $OWN$,NOEXE,2

00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 0108C FLAGS_VECTOR:
                                                                                .ADDRESS P.AEQ, P.AEP, P.AEY, P.AEO, P.AEV, -           ;
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 010A4                      P.AEW, P.AES, P.AER, P.AEX, P.AEZ, P.AFA, -     ;
                                        00000000' 00000000' 010BC                       P.AEN, P.AEU, P.AET                             ;
                                                            010C4               .BLKB   72                                              ;

                                                                    USERNAME=           P.ADB
                                                                    ACCOUNT=            P.ADC
                                                                    CLI_TABLE=          P.ADD
                                                                    DEFAULT=            P.ADE
                                                                    LGICMD=             P.ADF
                                                                    FLAGS=              P.ADG
                                                                    FLAG_PAD=           P.ADH
                                                                    PRIMDAYS=           P.ADI
                                                                    SECDAYS=            P.ADJ
                                                                    NORESTRICT=         P.ADK
                                                                    ACCESSHDR1=         P.ADL
                                                                    ACCESSHDR2=         P.ADM
                                                                    NETACCESS=          P.ADN
                                                                    BATACCESS=          P.ADO
                                                                    LOCACCESS=          P.ADP
                                                                    DIAACCESS=          P.ADQ
                                                                    REMACCESS=          P.ADR
                                                                    EXPIRATION=         P.ADS
                                                                    PWDDATA=            P.ADT
                                                                    LASTLOGIN=          P.ADU
                                                                    QUOTA1=             P.ADV
                                                                    QUOTA2=             P.ADW
                                                                    QUOTA3=             P.ADX
                                                                    QUOTA4=             P.ADY
                                                                    QUOTA5=             P.ADZ
                                                                    QUOTA6=             P.AEA
                                                                    QUOTA7=             P.AEB
                                                                    PRIVS=              P.AEC
                                                                    DEFPRIVS=           P.AED
                                                                    NULLSTR=            P.AEE
                                                                    MON=                P.AEF
                                                                    TUE=                P.AEG
                                                                    WED=                P.AEH
                                                                    THU=                P.AEI
                                                                    FRI=                P.AEJ
                                                                    SAT=                P.AEK
                                                                    SUN=                P.AEL
                                                                    NODAY=              P.AEM
                                                                    CPUTIME=            RECBUF+556

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

```
                                    0FFC 00000 DISPLAY_FULL:
                                                        .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11                    ; 3542
                                5E      FF1C    CE  9E 00002                MOVAB   -228(SP), SP
                                31 00000000'    00  E9 00007                BLBC    STR_WILD, 1$                            ; 3663
                                55 00000000'    00  9E 0000E                MOVAB   MATCH_TOKEN, R5                         ; 3664
                                53 00000000'    00  9E 00015                MOVAB   RECBUF+4, R3
00000000'  00                   20              20  3A 0001C                LOCC    #32, #32, RECBUF+4                      ; 3663
                                52          E0  A0  9E 00024                MOVAB   -32(R0), R2
                                52              52  CE 00028                MNEGL   R2, R2
                                54 00000000'    00  D0 0002B                MOVL    MATCH_TOKENLEN, R4                      ; 3664
                                   00000000G    00  16 00032                JSB     FMG$MATCH_NAME
                                04              50  E8 00038                BLBS    R0, 1$
                                50              01  D0 0003B                MOVL    #1, R0                                  ; 3666
                                                    04 0003E                RET
                    00000000'   00              01  D0 0003F 1$:           MOVL    #1, FOUND_MATCH                         ; 3668
                                50 00000000'    00  D0 00046                MOVL    RABPTR, R0
                                           22   A0  B4 0004D                CLRW    34(R0)
                                                50  DD 00050                PUSHL   R0
                    00000000G   00              01  FB 00052                CALLS   #1, SYS$PUT
                    00000000'   00 00000000'    00  9B 00059                MOVZBW  USERNAME, FAODSC                        ; 3674
                    00000000'   00 00000000'    00  9E 00064                MOVAB   USERNAME+1, FAODSC+4
                                   00000000'    00  9F 0006F                PUSHAB  RECBUF+84
                                   00000000'    00  9F 00075                PUSHAB  RECBUF+4
                                                20  DD 0007B                PUSHL   #32
                                   00000000'    00  9F 0007D                PUSHAB  DISDSC
                7E 00000000'    00              22  C1 00083                ADDL3   #34, RABPTR, -(SP)
                                   00000000'    00  9F 0008B                PUSHAB  FAODSC
                    00000000G   00              06  FB 00091                CALLS   #6, SYS$FAO
                                   00000000'    00  DD 00098                PUSHL   RABPTR
                    00000000G   00              01  FB 0009E                CALLS   #1, SYS$PUT
                    00000000'   00 00000000'    00  9B 000A5                MOVZBW  ACCOUNT, FAODSC                         ; 3679
                    00000000'   00 00000000'    00  9E 000B0                MOVAB   ACCOUNT+1, FAODSC+4
                                50 00000000'    00  D0 000BB                MOVL    RECBUF+36, R0
                                                50  DD 000C2                PUSHL   R0
                                                50  DD 000C4                PUSHL   R0
                                   00000000'    00  9F 000C6                PUSHAB  RECBUF+52
                                                20  DD 000CC                PUSHL   #32
                                   00000000'    00  9F 000CE                PUSHAB  DISDSC
                7E 00000000'    00              22  C1 000D4                ADDL3   #34, RABPTR, -(SP)
                                   00000000'    00  9F 000DC                PUSHAB  FAODSC
                    00000000G   00              07  FB 000E2                CALLS   #7, SYS$FAO
                                   00000000'    00  DD 000E9                PUSHL   RABPTR
                    00000000G   00              01  FB 000EF                CALLS   #1, SYS$PUT
                    00000000'   00 00000000'    00  9B 000F6                MOVZBW  CLI_TABLE, FAODSC                       ; 3683
                    00000000'   00 00000000'    00  9E 00101                MOVAB   CLI_TABLE+1, FAODSC+4
                                   00000000'    00  9F 0010C                PUSHAB  RECBUF+308
                                   00000000'    00  9F 00112                PUSHAB  RECBUF+276
                                   00000000'    00  9F 00118                PUSHAB  DISDSC
                7E 00000000'    00              22  C1 0011E                ADDL3   #34, RABPTR, -(SP)
                                   00000000'    00  9F 00126                PUSHAB  FAODSC
                    00000000G   00              05  FB 0012C                CALLS   #5, SYS$FAO
                                   00000000'    00  DD 00133                PUSHL   RABPTR
                    00000000G   00              01  FB 00139                CALLS   #1, SYS$PUT
                    00000000'   00 00000000'    00  9B 00140                MOVZBW  DEFAULT, FAODSC                         ; 3687
                    00000000'   00 00000000'    00  9E 0014B                MOVAB   DEFAULT+1, FAODSC+4
                                   00000000'    00  9F 00156                PUSHAB  RECBUF+148
                                   00000000'    00  9F 0015C                PUSHAB  RECBUF+116
```

```
                                00000000'  00  9F  00162          PUSHAB  DISDSC                                                    ;
                      7E 00000000'  00         22  C1  00168          ADDL3   #34, RABPTR, -(SP)                                    ;
                                00000000'  00  9F  00170          PUSHAB  FAODSC                                                   ;
                         00000000G  00         05  FB  00176          CALLS   #5, SYS$FAO                                          ;
                                00000000'  00  DD  0017D          PUSHL   RABPTR                                                   ;
                         00000000G  00         01  FB  00183          CALLS   #1, SYS$PUT                                          ;
                         00000000'  00 00000000'  00  9B  0018A          MOVZBW  LGICMD, FAODSC                                   ; 3690
                         00000000'  00 00000000'  00  9E  00195          MOVAB   LGICMD+1, FAODSC+4                               ;
                                00000000'  00  9F  001A0          PUSHAB  RECBUF+212                                               ;
                                00000000'  00  9F  001A6          PUSHAB  DISDSC                                                   ;
                      7E 00000000'  00         22  C1  001AC          ADDL3   #34, RABPTR, -(SP)                                   ;
                                00000000'  00  9F  001B4          PUSHAB  FAODSC                                                   ;
                         00000000G  00         04  FB  001BA          CALLS   #4, SYS$FAO                                          ;
                                00000000'  00  DD  001C1          PUSHL   RABPTR                                                   ;
                         00000000G  00         01  FB  001C7          CALLS   #1, SYS$PUT                                          ;
                                               57  D4  001CE          CLRL    COUNT                                            ; 3692
                                   5B 00000000'  00  9A  001D0          MOVZBL  FLAGS, R11                                       ; 3693
                                   56            5B  D0  001D7          MOVL    R11, LCOUNT                                      ;
                                               59  D4  001DA          CLRL    J                                                ; 3694
                      47 00000000'  00         59  E1  001DC 2$:       BBC     J, RECBUF+468, 4$                                ; 3697
                                   5A 00000000'0049  D0  001E4          MOVL    FLAGS_VECTOR[J], FLAG_STRING                     ; 3698
                                               3D  13  001EC          BEQL    4$                                               ;
                                   58            6A  9A  001EE          MOVZBL  (FLAG_STRING), R8                                ; 3701
                                   58            56  C0  001F1          ADDL2   LCOUNT, R8                                       ;
                            00000050   8F        58  D1  001F4          CMPL    R8, #80                                          ;
                                               18  1B  001FB          BLEQU   3$                                               ;
                                   58 00000000'  00  9A  001FD          MOVZBL  FLAG_PAD, R8                                     ; 3704
                   44 AE47 00000000'  00         58  28  00204          MOVC3   R8, FLAG_PAD+1, STRING[COUNT]                    ;
                                   57            58  C0  0020E          ADDL2   R8, COUNT                                        ; 3705
                                   56      FE    A8  9E  00211          MOVAB   -2(R8), LCOUNT                                   ; 3706
                                   50            6A  9A  00215 3$:       MOVZBL  (FLAG_STRING), R0                                ; 3708
                   44 AE47        01    AA           50  28  00218          MOVC3   R0, 1(FLAG_STRING), STRING[COUNT]                ;
                                   50            6A  9A  0021F          MOVZBL  (FLAG_STRING), R0                                ; 3709
                                   57            50  C0  00222          ADDL2   R0, COUNT                                        ;
                                   50            6A  9A  00225          MOVZBL  (FLAG_STRING), R0                                ; 3710
                                   56            50  C0  00228          ADDL2   R0, LCOUNT                                       ;
                         AD                      59  1F  F3  0022B 4$:       AOBLEQ  #31, J, 2$                               ; 3694
                            00000000'  00        5B  B0  0022F          MOVW    R11, FAODSC                                      ; 3715
                            00000000'  00 00000000'  00  9E  00236          MOVAB   FLAGS+1, FAODSC+4                                ;
                                              44  AE  9F  00241          PUSHAB  STRING                                           ;
                                               57  DD  00244          PUSHL   COUNT                                            ;
                                00000000'  00  9F  00246          PUSHAB  DISDSC                                                   ;
                      7E 00000000'  00         22  C1  0024C          ADDL3   #34, RABPTR, -(SP)                                   ;
                                00000000'  00  9F  00254          PUSHAB  FAODSC                                                   ;
                         00000000G  00         05  FB  0025A          CALLS   #5, SYS$FAO                                          ;
                                00000000'  00  DD  00261          PUSHL   RABPTR                                                   ;
                         00000000G  00         01  FB  00267          CALLS   #1, SYS$PUT                                          ;
                         00000000'  00 00000000'  00  9B  0026E          MOVZBW  PRIMDAYS, FAODSC                                 ; 3724
                         00000000'  00 00000000'  00  9E  00279          MOVAB   PRIMDAYS+1, FAODSC+4                             ;
                      09 00000000'  00         06  E1  00284          BBC     #6, RECBUF+514, 5$                               ;
                                   50 00000000'  00  9E  0028C          MOVAB   NODAY, R0                                        ;
                                               07  11  00293          BRB     6$                                               ;
                                   50 00000000'  00  9E  00295 5$:       MOVAB   SUN, R0                                          ;
                                               50  DD  0029C 6$:       PUSHL   R0                                               ;
                      09 00000000'  00         05  E1  0029E          BBC     #5, RECBUF+514, 7$                               ;
                                   50 00000000'  00  9E  002A6          MOVAB   NODAY, R0                                        ;
                                               07  11  002AD          BRB     8$                                               ;
```

```
                    50 00000000'   00  9E 002AF 7$:      MOVAB   SAT, R0
                                   50  DD 002B6 8$:      PUSHL   R0
                09 00000000'   00       04  E1 002B8          BBC     #4, RECBUF+514, 9$
                    50 00000000'   00  9E 002C0           MOVAB   NODAY, R0
                                   07  11 002C7          BRB     10$
                    50 00000000'   00  9E 002C9 9$:      MOVAB   FRI, R0
                                   50  DD 002D0 10$:     PUSHL   R0
                09 00000000'   00       03  E1 002D2          BBC     #3, RECBUF+514, 11$
                    50 00000000'   00  9E 002DA           MOVAB   NODAY, R0
                                   07  11 002E1          BRB     12$
                    50 00000000'   00  9E 002E3 11$:     MOVAB   THU, R0
                                   50  DD 002EA 12$:     PUSHL   R0
                09 00000000'   00       02  E1 002EC          BBC     #2, RECBUF+514, 13$
                    50 00000000'   00  9E 002F4           MOVAB   NODAY, R0
                                   07  11 002FB          BRB     14$
                    50 00000000'   00  9E 002FD 13$:     MOVAB   WED, R0
                                   50  DD 00304 14$:     PUSHL   R0
                09 00000000'   00       01  E1 00306          BBC     #1, RECBUF+514, 15$
                    50 00000000'   00  9E 0030E           MOVAB   NODAY, R0
                                   07  11 00315          BRB     16$
                    50 00000000'   00  9E 00317 15$:     MOVAB   TUE, R0
                                   50  DD 0031E 16$:     PUSHL   R0
                    09 00000000'   00  E9 00320          BLBC    RECBUF+514, 17$
                    50 00000000'   00  9E 00327           MOVAB   NODAY, R0
                                   07  11 0032E          BRB     18$
                    50 00000000'   00  9E 00330 17$:     MOVAB   MON, R0
                                   50  DD 00337 18$:     PUSHL   R0
                       00000000'   00  9F 00339          PUSHAB  DISDSC
            7E 00000000'   00       22  C1 0033F          ADDL3   #34, RABPTR, -(SP)
                       00000000'   00  9F 00347          PUSHAB  FAODSC
                00000000G  00       0A  FB 0034D          CALLS   #10, SYS$FAO
                       00000000'   00  DD 00354          PUSHL   RABPTR
                00000000G  00       01  FB 0035A          CALLS   #1, SYS$PUT
                00000000'  00  00000000'   00  9B 00361          MOVZBW  SECDAYS, FAODSC                 ; 3733
                00000000'  00  00000000'   00  9E 0036C          MOVAB   SECDAYS+1, FAODSC+4
            09 00000000'   00       06  E1 00377          BBC     #6, RECBUF+514, 19$
                    50 00000000'   00  9E 0037F           MOVAB   SUN, R0
                                   07  11 00386          BRB     20$
                    50 00000000'   00  9E 00388 19$:     MOVAB   NODAY, R0
                                   50  DD 0038F 20$:     PUSHL   R0
                09 00000000'   00       05  E1 00391          BBC     #5, RECBUF+514, 21$
                    50 00000000'   00  9E 00399           MOVAB   SAT, R0
                                   07  11 003A0          BRB     22$
                    50 00000000'   00  9E 003A2 21$:     MOVAB   NODAY, R0
                                   50  DD 003A9 22$:     PUSHL   R0
                09 00000000'   00       04  E1 003AB          BBC     #4, RECBUF+514, 23$
                    50 00000000'   00  9E 003B3           MOVAB   FRI, R0
                                   07  11 003BA          BRB     24$
                    50 00000000'   00  9E 003BC 23$:     MOVAB   NODAY, R0
                                   50  DD 003C3 24$:     PUSHL   R0
                09 00000000'   00       03  E1 003C5          BBC     #3, RECBUF+514, 25$
                    50 00000000'   00  9E 003CD           MOVAB   THU, R0
                                   07  11 003D4          BRB     26$
                    50 00000000'   00  9E 003D6 25$:     MOVAB   NODAY, R0
                                   50  DD 003DD 26$:     PUSHL   R0
                09 00000000'   00       02  E1 003DF          BBC     #2, RECBUF+514, 27$
                    50 00000000'   00  9E 003E7           MOVAB   WED, R0
```

```
                                                07  11  003EE                 BRB     28$                                                     ;
                                    50 00000000'00  9E  003F0 27$:           MOVAB   NODAY, R0                                               ;
                                                50  DD  003F7 28$:           PUSHL   R0                                                      ;
                                09 00000000'    00  01  E1  003F9           BBC     #1, RECBUF+514, 29$                                     ;
                                    50 00000000'00  9E  00401               MOVAB   TUE, R0                                                 ;
                                                07  11  00408               BRB     30$                                                     ;
                                    50 00000000'00  9E  0040A 29$:          MOVAB   NODAY, R0                                               ;
                                                50  DD  00411 30$:          PUSHL   R0                                                      ;
                                    09 00000000'00  E9  00413               BLBC    RECBUF+514, 31$                                         ;
                                    50 00000000'00  9E  0041A               MOVAB   MON, R0                                                 ;
                                                07  11  00421               BRB     32$                                                     ;
                                    50 00000000'00  9E  00423 31$:          MOVAB   NODAY, R0                                               ;
                                                50  DD  0042A 32$:          PUSHL   R0                                                      ;
                                    00000000'   00  9F  0042C               PUSHAB  DISDSC                                                  ;
                                7E 00000000'    00  22  C1  00432           ADDL3   #34, RABPTR, -(SP)                                      ;
                                    00000000'   00  9F  0043A               PUSHAB  FAODSC                                                  ;
                                   00000000G    00  0A  FB  00440           CALLS   #10, SYS$FAO                                            ;
                                    00000000'   00  DD  00447               PUSHL   RABPTR                                                  ;
                                   00000000G    00  01  FB  0044D           CALLS   #1, SYS$PUT                                             ;
                                52 00000000'    00  22  C1  00454           ADDL3   #34, RABPTR, R2                                         ; 3738
                    00000000'   00              1E  00  3B  0045C           SKPC    #0, #30, RECBUF+472                                     ; 3735
                                                02  12  00464               BNEQ    33$                                                     ;
                                                51  D4  00466               CLRL    R1                                                      ;
                                                51  D5  00468 33$:          TSTL    R1                                                      ;
                                                3B  12  0046A               BNEQ    34$                                                     ;
                                   00000000'    00 00000000'00  9B  0046C   MOVZBW  NORESTRICT, FAODSC                                      ; 3738
                                   00000000'    00 00000000'00  9E  00477   MOVAB   NORESTRICT+1, FAODSC+4                                  ;
                                    00000000'   00  9F  00482               PUSHAB  DISDSC                                                  ;
                                                52  DD  00488               PUSHL   R2                                                      ;
                                    00000000'   00  9F  0048A               PUSHAB  FAODSC                                                  ;
                                   00000000G    00  03  FB  00490           CALLS   #3, SYS$FAO                                             ;
                                    00000000'   00  DD  00497               PUSHL   RABPTR                                                  ;
                                   00000000G    00  01  FB  0049D           CALLS   #1, SYS$PUT                                             ;
                                              00D5  31  00  004A4           BRW     35$                                                     ; 3735
                                   00000000'    00 00000000'00  9B  004A7 34$:   MOVZBW  ACCESSHDR1, FAODSC                               ; 3742
                                   00000000'    00 00000000'00  9E  004B2   MOVAB   ACCESSHDR1+1, FAODSC+4                                 ;
                                    00000000'   00  9F  004BD               PUSHAB  DISDSC                                                  ;
                                                52  DD  004C3               PUSHL   R2                                                      ;
                                    00000000'   00  9F  004C5               PUSHAB  FAODSC                                                  ;
                                   00000000G    00  03  FB  004CB           CALLS   #3, SYS$FAO                                             ;
                                    00000000'   00  DD  004D2               PUSHL   RABPTR                                                  ;
                                   00000000G    00  01  FB  004D8           CALLS   #1, SYS$PUT                                             ;
                                   00000000'    00 00000000'00  9B  004DF   MOVZBW  ACCESSHDR2, FAODSC                                      ; 3743
                                   00000000'    00 00000000'00  9E  004EA   MOVAB   ACCESSHDR2+1, FAODSC+4                                  ;
                                    00000000'   00  9F  004F5               PUSHAB  DISDSC                                                  ;
                                7E 00000000'    00  22  C1  004FB           ADDL3   #34, RABPTR, -(SP)                                      ;
                                    00000000'   00  9F  00503               PUSHAB  FAODSC                                                  ;
                                   00000000G    00  03  FB  00509           CALLS   #3, SYS$FAO                                             ;
                                    00000000'   00  DD  00510               PUSHL   RABPTR                                                  ;
                                   00000000G    00  01  FB  00516           CALLS   #1, SYS$PUT                                             ;
                                    00000000'   00  9F  0051D               PUSHAB  RECBUF+472                                              ; 3745
                                    00000000'   00  9F  00523               PUSHAB  NETACCESS                                               ;
                                   00000000V    00  02  FB  00529           CALLS   #2, DISPLAY_HOURS                                       ;
                                    00000000'   00  9F  00530               PUSHAB  RECBUF+478                                              ; 3746
                                    00000000'   00  9F  00536               PUSHAB  BATACCESS                                               ;
                                   00000000V    00  02  FB  0053C           CALLS   #2, DISPLAY_HOURS                                       ;
                                    00000000'   00  9F  00543               PUSHAB  RECBUF+484                                              ; 3747
```

```
                                00000000'  00   9F 00549          PUSHAB  LOCACCESS
                    00000000V   00            02   FB 0054F          CALLS   #2, DISPLAY_HOURS
                                00000000'  00   9F 00556          PUSHAB  RECBUF+490                      ;3748
                                00000000'  00   9F 0055C          PUSHAB  DIAACCESS
                    00000000V   00            02   FB 00562          CALLS   #2, DISPLAY_HOURS
                                00000000'  00   9F 00569          PUSHAB  RECBUF+496                      ;3749
                                00000000'  00   9F 0056F          PUSHAB  REMACCESS
                    00000000V   00            02   FB 00575          CALLS   #2, DISPLAY_HOURS
                                      28   AE   9F 0057C 35$:       PUSHAB  TIME1                           ;3752
                                           11   DD 0057F          PUSHL   #17
                                00000000'  00   9F 00581          PUSHAB  RECBUF+364
                    00000000V   00            03   FB 00587          CALLS   #3, CONVERT_TIME
                    00000000'   00 00000000'  00   9B 0058E          MOVZBW  EXPIRATION, FAODSC              ;3756
                    00000000'   00 00000000'  00   9E 00599          MOVAB   EXPIRATION+1, FAODSC+4
                                7E 00000000'  00   3C 005A4          MOVZWL  RECBUF+356, -(SP)
                                7E 00000000'  00   9A 005AB          MOVZBL  RECBUF+362, -(SP)
                                      30   AE   9F 005B2          PUSHAB  TIME1
                                           11   DD 005B5          PUSHL   #17
                                00000000'  00   9F 005B7          PUSHAB  DISDSC
                7E 00000000'  00            22   C1 005BD          ADDL3   #34, RABPTR, -(SP)
                                00000000'  00   9F 005C5          PUSHAB  FAODSC
                    00000000G   00            07   FB 005CB          CALLS   #7, SYS$FAO
                                00000000'  00   DD 005D2          PUSHL   RABPTR
                    00000000G   00            01   FB 005D8          CALLS   #1, SYS$PUT
                                      28   AE   9F 005DF          PUSHAB  TIME1                           ;3758
                                           0A   DD 005E2          PUSHL   #10
                                00000000'  00   9F 005E4          PUSHAB  RECBUF+372
                    00000000V   00            03   FB 005EA          CALLS   #3, CONVERT_TIME
                                50 00000000'  00   9E 005F1          MOVAB   RECBUF+380, PTR                 ;3759
                    FFFFFFFF    8F            60   D1 005F8          CMPL    (PTR), #-1                      ;3760
                                           15   12 005FF          BNEQ    36$
                    FFFFFFFF    8F      04    A0   D1 00601          CMPL    4(PTR), #-1
                                           0B   12 00609          BNEQ    36$
             14  AE 00000000'  00            11   28 0060B          MOVC3   #17, P.AFB, TIME2               ;3761
                                           0E   11 00614          BRB     37$
                                      14   AE   9F 00616 36$:       PUSHAB  TIME2                           ;3763
                                           11   DD 00619          PUSHL   #17
                                           50   DD 0061B          PUSHL   PTR
                    00000000V   00            03   FB 0061D          CALLS   #3, CONVERT_TIME
                                           5E   DD 00624 37$:       PUSHL   SP                              ;3764
                                           11   DD 00626          PUSHL   #17
                                00000000'  00   9F 00628          PUSHAB  RECBUF+388
                    00000000V   00            03   FB 0062E          CALLS   #3, CONVERT_TIME
                    00000000'   00 00000000'  00   9B 00635          MOVZBW  PWDDATA, FAODSC                 ;3770
                    00000000'   00 00000000'  00   9E 00640          MOVAB   PWDDATA+1, FAODSC+4
                                           5E   DD 0064B          PUSHL   SP
                50 00000000'  00 00000000'  00   C9 0064D          BISL3   RECBUF+392, RECBUF+388, R0
                                           04   12 00659          BNEQ    38$
                                           7E   D4 0065B          CLRL    -(SP)
                                           02   11 0065D          BRB     39$
                                           11   DD 0065F 38$:       PUSHL   #17
                                      1C   AE   9F 00661 39$:       PUSHAB  TIME2
                                           11   DD 00664          PUSHL   #17
                                      38   AE   9F 00666          PUSHAB  TIME1
                                           0A   DD 00669          PUSHL   #10
                                00000000'  00   9F 0066B          PUSHAB  DISDSC
                7E 00000000'  00            22   C1 00671          ADDL3   #34, RABPTR, -(SP)
```

```
                                00000000'  00  9F  00679          PUSHAB  FAODSC                                          ;
                        00000000G  00          09  FB  0067F          CALLS   #9, SYS$FAO                                     ;
                                00000000'  00  DD  00686          PUSHL   RABPTR                                          ;
                        00000000G  00          01  FB  0068C          CALLS   #1, SYS$PUT                                     ;
                                      28   AE  9F  00693          PUSHAB  TIME1                                           ; 3772
                                           11  DD  00696          PUSHL   #17                                             ;
                                00000000'  00  9F  00698          PUSHAB  RECBUF+396                                      ;
                        00000000V  00          03  FB  0069E          CALLS   #3, CONVERT_TIME                                ;
                                      14   AE  9F  006A5          PUSHAB  TIME2                                           ; 3773
                                           11  DD  006A8          PUSHL   #17                                             ;
                                00000000'  00  9F  006AA          PUSHAB  RECBUF+404                                      ;
                        00000000V  00          03  FB  006B0          CALLS   #3, CONVERT_TIME                                ;
                        00000000'  00  00000000'  00  9B  006B7          MOVZBW  LASTLOGIN, FAODSC                               ; 3776
                        00000000'  00  00000000'  00  9E  006C2          MOVAB   LASTLOGIN+1, FAODSC+4                           ;
                                      14   AE  9F  006CD          PUSHAB  TIME2                                           ;
                                           11  DD  006D0          PUSHL   #17                                             ;
                                      30   AE  9F  006D2          PUSHAB  TIME1                                           ;
                                           11  DD  006D5          PUSHL   #17                                             ;
                                00000000'  00  9F  006D7          PUSHAB  DISDSC                                          ;
                     7E 00000000'  00          22  C1  006DD          ADDL3   #34, RABPTR, -(SP)                              ;
                                00000000'  00  9F  006E5          PUSHAB  FAODSC                                          ;
                        00000000G  00          07  FB  006EB          CALLS   #7, SYS$FAO                                     ;
                                00000000'  00  DD  006F2          PUSHL   RABPTR                                          ;
                        00000000G  00          01  FB  006F8          CALLS   #1, SYS$PUT                                     ;
              51 00000000'  00            1F  01  EF  006FF          EXTZV   #1, #31, CPUTIME, R1                            ; 3778
                                09 00000000'  00  E9  00708          BLBC    CPUTIME, 40$                                    ; 3779
                                50 FFFE7960   8F  D0  0070F          MOVL    #-100000, R0                                    ;
                                              02  11  00716          BRB     41$                                             ;
                                              50  D4  00718  40$:     CLRL    R0                                              ;
   3C   AE              50            51 FFFCF2C0   8F  7A  0071A  41$:     EMUL    #-200000, R1, R0, DELTA_TIME                    ; 3778
                                      28   AE  9F  00724          PUSHAB  TIME1                                           ; 3780
                                           0D  DD  00727          PUSHL   #13                                             ;
                                      44   AE  9F  00729          PUSHAB  DELTA_TIME                                      ;
                        00000000V  00          03  FB  0072C          CALLS   #3, CONVERT_TIME                                ;
                        00000000'  00  00000000'  00  9B  00733          MOVZBW  QUOTA1, FAODSC                                  ; 3785
                        00000000'  00  00000000'  00  9E  0073E          MOVAB   QUOTA1+1, FAODSC+4                              ;
                                00000000'  00  DD  00749          PUSHL   RECBUF+560                                      ;
                                7E 00000000'  00  3C  0074F          MOVZWL  RECBUF+536, -(SP)                               ;
                                7E 00000000'  00  3C  00756          MOVZWL  RECBUF+518, -(SP)                               ;
                                00000000'  00  9F  0075D          PUSHAB  DISDSC                                          ;
                     7E 00000000'  00          22  C1  00763          ADDL3   #34, RABPTR, -(SP)                              ;
                                00000000'  00  9F  0076B          PUSHAB  FAODSC                                          ;
                        00000000G  00          06  FB  00771          CALLS   #6, SYS$FAO                                     ;
                                00000000'  00  DD  00778          PUSHL   RABPTR                                          ;
                        00000000G  00          01  FB  0077E          CALLS   #1, SYS$PUT                                     ;
                        00000000'  00  00000000'  00  9B  00785          MOVZBW  QUOTA2, FAODSC                                  ; 3790
                        00000000'  00  00000000'  00  9E  00790          MOVAB   QUOTA2+1, FAODSC+4                              ;
                                00000000'  00  DD  00798          PUSHL   RECBUF+564                                      ;
                                7E 00000000'  00  3C  007A1          MOVZWL  RECBUF+538, -(SP)                               ;
                                7E 00000000'  00  3C  007A8          MOVZWL  RECBUF+520, -(SP)                               ;
                                00000000'  00  9F  007AF          PUSHAB  DISDSC                                          ;
                     7E 00000000'  00          22  C1  007B5          ADDL3   #34, RABPTR, -(SP)                              ;
                                00000000'  00  9F  007BD          PUSHAB  FAODSC                                          ;
                        00000000G  00          06  FB  007C3          CALLS   #6, SYS$FAO                                     ;
                                00000000'  00  DD  007CA          PUSHL   RABPTR                                          ;
                        00000000G  00          01  FB  007D0          CALLS   #1, SYS$PUT                                     ;
                        00000000'  00  00000000'  00  9B  007D7          MOVZBW  QUOTA3, FAODSC                                  ; 3795
```

```
                00000000'  00 00000000'  00  9E 007E2         MOVAB   QUOTA3+1, FAODSC+4                                ;
                           00000000'  00  DD 007ED         PUSHL   RECBUF+568                                        ;
                        7E 00000000'  00  3C 007F3         MOVZWL  RECBUF+526, -(SP)                                 ;
                        7E 00000000'  00  3C 007FA         MOVZWL  RECBUF+522, -(SP)                                 ;
                           00000000'  00  9F 00801         PUSHAB  DISDSC                                            ;
             7E 00000000'  00             22  C1 00807         ADDL3   #34, RABPTR, -(SP)                                ;
                           00000000'  00  9F 0080F         PUSHAB  FAODSC                                            ;
                00000000G  00             06  FB 00815         CALLS   #6, SYS$FAO                                       ;
                           00000000'  00  DD 0081C         PUSHL   RABPTR                                            ;
                00000000G  00             01  FB 00822         CALLS   #1, SYS$PUT                                       ;
                00000000'  00 00000000'  00  9B 00829         MOVZBW  QUOTA4, FAODSC                                    ; 3800
                00000000'  00 00000000'  00  9E 00834         MOVAB   QUOTA4+1, FAODSC+4                                ;
                           00000000'  00  DD 0083F         PUSHL   RECBUF+544                                        ;
                        7E 00000000'  00  3C 00845         MOVZWL  RECBUF+528, -(SP)                                 ;
                        7E 00000000'  00  3C 0084C         MOVZWL  RECBUF+524, -(SP)                                 ;
                           00000000'  00  9F 00853         PUSHAB  DISDSC                                            ;
             7E 00000000'  00             22  C1 00859         ADDL3   #34, RABPTR, -(SP)                                ;
                           00000000'  00  9F 00861         PUSHAB  FAODSC                                            ;
                00000000G  00             06  FB 00867         CALLS   #6, SYS$FAO                                       ;
                           00000000'  00  DD 0086E         PUSHL   RABPTR                                            ;
                00000000G  00             01  FB 00874         CALLS   #1, SYS$PUT                                       ;
                00000000'  00 00000000'  00  9B 0087B         MOVZBW  QUOTA5, FAODSC                                    ; 3805
                00000000'  00 00000000'  00  9E 00886         MOVAB   QUOTA5+1, FAODSC+4                                ;
                           00000000'  00  DD 00891         PUSHL   RECBUF+540                                        ;
                        7E 00000000'  00  3C 00897         MOVZWL  RECBUF+532, -(SP)                                 ;
                        7E 00000000'  00  9A 0089E         MOVZBL  RECBUF+516, -(SP)                                 ;
                           00000000'  00  9F 008A5         PUSHAB  DISDSC                                            ;
             7E 00000000'  00             22  C1 008AB         ADDL3   #34, RABPTR, -(SP)                                ;
                           00000000'  00  9F 008B3         PUSHAB  FAODSC                                            ;
                00000000G  00             06  FB 008B9         CALLS   #6, SYS$FAO                                       ;
                           00000000'  00  DD 008C0         PUSHL   RABPTR                                            ;
                00000000G  00             01  FB 008C6         CALLS   #1, SYS$PUT                                       ;
                00000000'  00 00000000'  00  9B 008CD         MOVZBW  QUOTA6, FAODSC                                    ; 3810
                00000000'  00 00000000'  00  9E 008D8         MOVAB   QUOTA6+1, FAODSC+4                                ;
                           00000000'  00  DD 008E3         PUSHL   RECBUF+548                                        ;
                        7E 00000000'  00  3C 008E9         MOVZWL  RECBUF+530, -(SP)                                 ;
                        7E 00000000'  00  9A 008F0         MOVZBL  RECBUF+517, -(SP)                                 ;
                           00000000'  00  9F 008F7         PUSHAB  DISDSC                                            ;
          7E 00000000'  00             22  C1 008FD         ADDL3   #34, RABPTR, -(SP)                                ;
                           00000000'  00  9F 00905         PUSHAB  FAODSC                                            ;
                00000000G  00             06  FB 00908         CALLS   #6, SYS$FAO                                       ;
                           00000000'  00  DD 00912         PUSHL   RABPTR                                            ;
                00000000G  00             01  FB 00918         CALLS   #1, SYS$PUT                                       ;
                00000000'  00 00000000'  00  9B 0091F         MOVZBW  QUOTA7, FAODSC                                    ; 3815
                00000000'  00 00000000'  00  9E 0092A         MOVAB   QUOTA7+1, FAODSC+4                                ;
                           00000000'  00  DD 00935         PUSHL   RECBUF+552                                        ;
                        7E 00000000'  00  3C 0093B         MOVZWL  RECBUF+534, -(SP)                                 ;
                                 30   AE  9F 00942         PUSHAB  TIME1                                             ;
                                      0D  DD 00945         PUSHL   #13                                               ;
                           00000000'  00  9F 00947         PUSHAB  DISDSC                                            ;
          7E 00000000'  00             22  C1 0094D         ADDL3   #34, RABPTR, -(SP)                                ;
                           00000000'  00  9F 00955         PUSHAB  FAODSC                                            ;
                00000000G  00             07  FB 0095B         CALLS   #7, SYS$FAO                                       ;
                           00000000'  00  DD 00962         PUSHL   RABPTR                                            ;
                00000000G  00             01  FB 00968         CALLS   #1, SYS$PUT                                       ;
                00000000'  00 00000000'  00  9B 0096F         MOVZBW  PRIVS, FAODSC                                     ; 3817
                00000000'  00 00000000'  00  9E 0097A         MOVAB   PRIVS+1, FAODSC+4                                 ;
```

```
                                00000000'  00  9F  00985          PUSHAB  DISDSC                                          ;
                    7E 00000000'  00       22  C1  00988          ADDL3   #34, RABPTR, -(SP)                             ;
                                00000000'  00  9F  00993          PUSHAB  FAODSC                                         ;
                       00000000G  00       03  FB  00999          CALLS   #3, SYS$FAO                                    ;
                                00000000'  00  DD  009A0          PUSHL   RABPTR                                         ;
                       00000000G  00       01  FB  009A6          CALLS   #1, SYS$PUT                                    ;
                                00000000'  00  9F  009AD          PUSHAB  RECBUF+412                                     ; 3818
                       00000000V  00       01  FB  009B3          CALLS   #1, PRINT_PRIV                                 ;
                       00000000'  00 00000000' 00  9B  009BA      MOVZBW  DEFPRIVS, FAODSC                               ; 3820
                       00000000'  00 00000000' 00  9E  009C5      MOVAB   DEFPRIVS+1, FAODSC+4                           ;
                                00000000'  00  9F  009D0          PUSHAB  DISDSC                                         ;
                    7E 00000000'  00       22  C1  009D6          ADDL3   #34, RABPTR, -(SP)                             ;
                                00000000'  00  9F  009DE          PUSHAB  FAODSC                                         ;
                       00000000G  00       03  FB  009E4          CALLS   #3, SYS$FAO                                    ;
                                00000000'  00  DD  009EB          PUSHL   RABPTR                                         ;
                       00000000G  00       01  FB  009F1          CALLS   #1, SYS$PUT                                    ;
                                00000000'  00  9F  009F8          PUSHAB  RECBUF+420                                     ; 3821
                       00000000V  00       01  FB  009FE          CALLS   #1, PRINT_PRIV                                 ;
                                1B 00000000'  00  E9  00A05      BLBC    RDB_EXIST$, 42$                                ; 3827
                       00000000G  00       00  FB  00A0C          CALLS   #0, UAF$BUILD_HOLDER                           ; 3830
                       00000000G  00       01  90  00A13          MOVB    #1, RDB_HEADER_FLAG                            ; 3831
                                00000000G  00  9F  00A1A          PUSHAB  HOLDER                                         ; 3832
                       00000000G  00       01  FB  00A20          CALLS   #1, UAF$WRITE_RIGHTS                           ;
                                           04  00A27  42$:        RET                                                    ; 3836

; Routine Size.  2600 bytes,    Routine Base:  $CODE$ + 16DB
```

```
;  3775     3837  1 %sbttl 'display_hours - display hourly restrictions'
;  3776     3838  1 routine display_hours (format_string, hour_vector) : novalue =
;  3777     3839  2 begin
;  3778     3840  2
;  3779     3841  2 !++
;  3780     3842  2 !
;  3781     3843  2 ! FUNCTIONAL DESCRIPTION:
;  3782     3844  2 !
;  3783     3845  2 !       Displays the hourly access for primary and secondary days
;  3784     3846  2 !       for one access type.
;  3785     3847  2 !
;  3786     3848  2 ! INPUTS:
;  3787     3849  2 !
;  3788     3850  2 !       format_string: address of FAO string for display
;  3789     3851  2 !       hour_vector: address of UAF record hourly vector
;  3790     3852  2 !       RABPTR - RMS data structure for the file
;  3791     3853  2 !
;  3792     3854  2 ! IMPLICIT INPUTS:
;  3793     3855  2 !
;  3794     3856  2 !       none
;  3795     3857  2 !
;  3796     3858  2 ! OUTPUTS:
;  3797     3859  2 !
;  3798     3860  2 !       none
;  3799     3861  2 !
;  3800     3862  2 ! IMPLICIT OUTPUTS:
;  3801     3863  2 !
;  3802     3864  2 !       none
;  3803     3865  2 !
;  3804     3866  2 ! ROUTINE VALUE:
;  3805     3867  2 !
;  3806     3868  2 !       none
;  3807     3869  2 !
;  3808     3870  2 ! SIDE EFFECTS:
;  3809     3871  2 !
;  3810     3872  2 !       none
;  3811     3873  2 !--
;  3812     3874  2
;  3813     3875  2 !
;  3814     3876  2 ! Display strings.
;  3815     3877  2 !
;  3816     3878  2
;  3817     3879  2 map
;  3818     3880  2     hour_vector : ref bitvector;
;  3819     3881  2
;  3820     3882  2 literal
;  3821     3883  2         yeschar         = '#',
;  3822     3884  2         nochar          = '-';
;  3823     3885  2
;  3824     3886  2 bind
;  3825     3887  2         noaccess        = cstring ('------  No access  -------'),
;  3826     3888  2         fullaccess      = cstring ('###### Full access #######');
;  3827     3889  2
;  3828     3890  2 local
;  3829     3891  2         pri_access      : vector [24, byte],    ! Character string for primary access
;  3830     3892  2         sec_access      : vector [24, byte];    ! Character string for secondary access
;  3831     3893  2
```

```
;  3832          3894  2 if . (hour_vector[0])<0,24> eql 0
;  3833          3895  2 then ch$move (24, fullaccess+1, pri_access)
;  3834          3896  2 else if . (hour_vector[0])<0,24> eql %x'ffffff'
;  3835          3897  2 then ch$move (24, noaccess+1, pri_access)
;  3836          3898  2 else incr j from 0 to 23
;  3837          3899  2 do
;  3838          3900  2     begin
;  3839          3901  2     if .hour_vector[.j]
;  3840          3902  2     then pri_access[.j] = nochar
;  3841          3903  2     else pri_access[.j] = yeschar;
;  3842          3904  2     end;
;  3843          3905  2
;  3844          3906  2 if . (hour_vector[0])<24,24> eql 0
;  3845          3907  2 then ch$move (24, fullaccess+1, sec_access)
;  3846          3908  2 else if . (hour_vector[0])<24,24> eql %x'ffffff'
;  3847          3909  2 then ch$move (24, noaccess+1, sec_access)
;  3848          3910  2 else incr j from 0 to 23
;  3849          3911  2 do
;  3850          3912  2     begin
;  3851          3913  2     if .hour_vector[.j+24]
;  3852          3914  2     then sec_access[.j] = nochar
;  3853          3915  2     else sec_access[.j] = yeschar;
;  3854          3916  2     end;
;  3855          3917  2
;  3856        P 3918  2 faomac (.format_string,
;  3857        P 3919  2         24, pri_access,
;  3858        P 3920  2         24, sec_access
;  3859          3921  2 );
;  3860          3922  1 end;


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                 18  00834 P.AFC:  .BYTE   24
73  65  63  63  61  20  6F  4E  20  20  2D  2D  2D  2D  2D  00835          .ASCII  \-----  No access  ------\                ;
                         2D  2D  2D  2D  2D  2D  20  20  73  00844
                                                 18  0084D P.AFD:  .BYTE   24
65  63  63  61  20  6C  6C  75  46  20  23  23  23  23  23  0084E          .ASCII  \##### Full access ######\               ;
                         23  23  23  23  23  23  20  73  73  0085D

                                                 NOACCESS=          P.AFC
                                                 FULLACCESS=        P.AFD

                                                        .PSECT  $CODE$,NOWRT,2

                                                 01FC 00000 DISPLAY_HOURS:
                                                            .WORD   Save R2,R3,R4,R5,R6,R7,R8                           ; 3838
                                              58 00000000'  00  9E 00002    MOVAB   FULLACCESS+1, R8                      ;
                                              57 00000000'  00  9E 00009    MOVAB   FAODSC, R7                            ;
                                              5E            30  C2 00010    SUBL2   #48, SP                               ;
                                              56      08    AC  D0 00013    MOVL    HOUR_VECTOR, R6                       ; 3894
                    00          66          18            00  ED 00017    CMPZV   #0, #24, (R6), #0                     ;
                                              07  12 0001C    BNEQ    1$                                            ;
                        18  AE          68                  18  28 0001E    MOVC3   #24, FULLACCESS+1, PRI_ACCESS         ; 3895
                                                            29  11 00023    BRB     6$                                    ;
```

```
00FFFFFF 8F                66              18                00 ED 00025 1$:    CMPZV   #0, #24, (R6), #16777215                ; 3896
                                                             08 12 00029        BNEQ    2$
                        18 AE          E7   A8                18 28 00030        MOVC3   #24, NOACCESS+1, PRI_ACCESS          ; 3897
                                                             16 11 00036        BRB     6$
                                                             50 D4 00038 2$:    CLRL    J                                    ; 3898
                           07                    66          50 E1 0003A 3$:    BBC     J, (R6), 4$                          ; 3901
                                          18 AE40             2D 90 0003E        MOVB    #45, PRI_ACCESS[J]                   ; 3902
                                                             05 11 00043        BRB     5$
                                          18 AE40             23 90 00045 4$:    MOVB    #35, PRI_ACCESS[J]                   ; 3903
                           EC                    50          17 F3 0004A 5$:    AOBLEQ  #23, J, 3$                           ; 3898
                     50    66                    18          18 EF 0004E 6$:    EXTZV   #24, #24, (R6), R0                   ; 3906
                                                             06 12 00053        BNEQ    7$
                           6E                    68          18 28 00055        MOVC3   #24, FULLACCESS+1, SEC_ACCESS        ; 3907
                                                             28 11 00059        BRB     12$
                              00FFFFFF           8F          50 D1 0005B 7$:    CMPL    R0, #16777215                        ; 3908
                                                             07 12 00062        BNEQ    8$
                           6E          E7       A8          18 28 00064        MOVC3   #24, NOACCESS+1, SEC_ACCESS          ; 3909
                                                             18 11 00069        BRB     12$
                                                             50 D4 0006B 8$:    CLRL    J                                    ; 3910
                                          51                18 A0 9E 0006D 9$: MOVAB   24(R0), R1                           ; 3913
                           06                    66          51 E1 00071        BBC     R1, (R6), 10$
                                       6E40                 2D 90 00075        MOVB    #45, SEC_ACCESS[J]                   ; 3914
                                                             04 11 00079        BRB     11$
                                       6E40                 23 90 0007B 10$:   MOVB    #35, SEC_ACCESS[J]                   ; 3915
                           EA                    50          17 F3 0007F 11$:   AOBLEQ  #23, J, 9$                           ; 3910
                                                 67          BC 9B 00083 12$:   MOVZBW  @FORMAT_STRING, FAODSC               ; 3921
                     04    A7          04       AC          01 C1 00087        ADDL3   #1, FORMAT_STRING, FAODSC+4
                                                             5E DD 0008D        PUSHL   SP
                                                             18 DD 0008F        PUSHL   #24
                                                          20 AE 9F 00091        PUSHAB  PRI_ACCESS
                                                             18 DD 00094        PUSHL   #24
                                                          F8 A7 9F 00096        PUSHAB  DISDSC
                        7E          08       A7              22 C1 00099        ADDL3   #34, RABPTR, -(SP)
                                                             57 DD 0009E        PUSHL   R7
                              00000000G  00                  07 FB 000A0        CALLS   #7, SYS$FAO
                                                          08 A7 DD 000A7        PUSHL   RABPTR
                              00000000G  00                  01 FB 000AA        CALLS   #1, SYS$PUT
                                                             04 000B1           RET                                          ; 3922

; Routine Size:  178 bytes,    Routine Base:  $CODE$ + 2103
```

```
;  3862     3923  1 %sbttl 'convert_time - convert time value to string'
;  3863     3924  1 routine convert_time (time, length, buffer) : novalue =
;  3864     3925  2 begin
;  3865     3926  2
;  3866     3927  2 !++
;  3867     3928  2 !
;  3868     3929  2 ! FUNCTIONAL DESCRIPTION:
;  3869     3930  2 !
;  3870     3931  2 !       Converts the binary time value into a string, substituting
;  3871     3932  2 !       the string " (none)" if the value is zero.
;  3872     3933  2 !
;  3873     3934  2 ! INPUTS:
;  3874     3935  2 !
;  3875     3936  2 !       time: address of quadword time
;  3876     3937  2 !       length: length of output string
;  3877     3938  2 !       buffer: address of output buffer
;  3878     3939  2 !
;  3879     3940  2 ! IMPLICIT INPUTS:
;  3880     3941  2 !
;  3881     3942  2 !       none
;  3882     3943  2 !
;  3883     3944  2 ! OUTPUTS:
;  3884     3945  2 !
;  3885     3946  2 !       none
;  3886     3947  2 !
;  3887     3948  2 ! IMPLICIT OUTPUTS:
;  3888     3949  2 !
;  3889     3950  2 !       none
;  3890     3951  2 !
;  3891     3952  2 ! ROUTINE VALUE:
;  3892     3953  2 !
;  3893     3954  2 !       none
;  3894     3955  2 !
;  3895     3956  2 ! SIDE EFFECTS:
;  3896     3957  2 !
;  3897     3958  2 !       none
;  3898     3959  2 !--
;  3899     3960  2
;  3900     3961  2 map
;  3901     3962  2         time            : ref vector;
;  3902     3963  2
;  3903     3964  2 local
;  3904     3965  2         string_desc     : vector [2];   ! descriptor for buffer
;  3905     3966  2
;  3906     3967  2 if (.time[0] or .time[1]) eql 0
;  3907     3968  2 then
;  3908     3969  2     begin
;  3909     3970  2     ch$fill (' ', .length-6, .buffer);
;  3910     3971  2     ch$move (6, uplit byte ('(none)'), .buffer+.length-6);
;  3911     3972  2     end
;  3912     3973  2
;  3913     3974  2 else
;  3914     3975  2     begin
;  3915     3976  2     string_desc[0] = .length;
;  3916     3977  2     string_desc[1] = .buffer;
;  3917     3978  2     $asctim (timadr = .time, timbuf = string_desc);
;  3918     3979  2     end;
```

```
; 3919          3980  1 end;


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                29  65  6E  6F  6E  28  00866 P.AFE:  .ASCII  \(none)\                                    ;

                                                                .EXTRN  SYS$ASCTIM

                                                                .PSECT  $CODE$,NOWRT,2

                                                        007C 00000 CONVERT_TIME:
                                                                .WORD   Save R2,R3,R4,R5,R6                             ; 3924
                                    5E                  08  C2 00002          SUBL2   #8, SP                          ;
                                    56              04  AC  D0 00005          MOVL    TIME, R6                        ; 3967
                                50  66              04  A6  C9 00009          BISL3   4(R6), (R6), R0                 ;
                                                        1C  12 0000E          BNEQ    1$                              ;
                                50          08      AC      06  C3 00010      SUBL3   #6, LENGTH, R0                  ; 3970
                            50  20          6E          00  2C 00015          MOVC5   #0, (SP), #32, R0, @BUFFER      ;
                                                    0C  BC      0001A         ;                                       ;
                                50          0C      AC  08  AC  C1 0001C      ADDL3   LENGTH, BUFFER, R0              ; 3971
                            FA  A0 00000000'        00      06  28 00022      MOVC3   #6, P.AFE, -6(R0)               ;
                                                            04  0002B          RET                                     ; 3967
                                    6E              08  AC  7D  0002C 1$:     MOVQ    LENGTH, STRING_DESC             ; 3976
                                                        7E  D4 00030          CLRL    -(SP)                           ; 3978
                                                        56  DD 00032          PUSHL   R6                              ;
                                                    08  AE  9F 00034          PUSHAB  STRING_DESC                     ;
                                                        7E  D4 00037          CLRL    -(SP)                           ;
                                00000000G       00          04  FB 00039      CALLS   #4, SYS$ASCTIM                  ;
                                                            04  00040          RET                                     ; 3980

; Routine Size:  65 bytes,    Routine Base:  $CODE$ + 21B5
```

```
;  3921     3981  1 %sbttl 'print_priv - print privilege bits'
;  3922     3982  1 routine print_priv (prvadr) : novalue =
;  3923     3983  2 begin
;  3924     3984  2
;  3925     3985  2 !++
;  3926     3986  2 !
;  3927     3987  2 ! FUNCTIONAL DESCRIPTION:
;  3928     3988  2 !
;  3929     3989  2 !       Routine to output the names of the privilege bits set
;  3930     3990  2 !       in the privilege vector supplied.
;  3931     3991  2 !
;  3932     3992  2 ! INPUTS:
;  3933     3993  2 !
;  3934     3994  2 !       PRVADR - Address of the privilege vector
;  3935     3995  2 !
;  3936     3996  2 ! IMPLICIT INPUTS:
;  3937     3997  2 !
;  3938     3998  2 !       PRV$AB_NAMES - table of privilege names and bit numbers
;  3939     3999  2 !
;  3940     4000  2 ! OUTPUTS:
;  3941     4001  2 !
;  3942     4002  2 !       none
;  3943     4003  2 !
;  3944     4004  2 ! IMPLICIT OUTPUTS:
;  3945     4005  2 !
;  3946     4006  2 !       none
;  3947     4007  2 !
;  3948     4008  2 ! ROUTINE VALUE:
;  3949     4009  2 !
;  3950     4010  2 !       none
;  3951     4011  2 !
;  3952     4012  2 ! SIDE EFFECTS:
;  3953     4013  2 !
;  3954     4014  2 !       none
;  3955     4015  2 !
;  3956     4016  2 !--
;  3957     4017  2
;  3958     4018  2 local
;  3959     4019  2         pointer,                        ! current location in PRV$AB_NAMES
;  3960     4020  2         prvcnt,                         ! number of names in DISBUF
;  3961     4021  2         symlen,                         ! length of bit name string
;  3962     4022  2         symmin,                         ! minimum symbol length
;  3963     4023  2         symval;                         ! value (bit number)
;  3964     4024  2
;  3965     4025  2 !
;  3966     4026  2 ! Initialize the buffer.
;  3967     4027  2 !
;  3968     4028  2
;  3969     4029  2 disbuf = ' ';                           ! insert blank at start
;  3970     4030  2 prvcnt = 0;
;  3971     4031  2 rabptr[rab$w_rsz] = 1;
;  3972     4032  2 pointer = prv$ab_names;                 ! point to symbol name table
;  3973     4033  2
;  3974     4034  2 while (symmin = . (.pointer)<0,8>) neq 0 ! pick up min symbol size
;  3975     4035  2 do
;  3976     4036  3     begin
;  3977     4037  3
```

```
;  3978          4038  3 !
;  3979          4039  3 ! Pick up the next bit name and number.  If the bit is set, insert
;  3980          4040  3 ! the bit name into the buffer.   When the buffer fills up output them
;  3981          4041  3 ! to the user.
;  3982          4042  3 !
;  3983          4043  3
;  3984          4044  3     pointer = .pointer + 1;
;  3985          4045  3     symval = . (.pointer)<0,8>;          ! get bit number
;  3986          4046  3     pointer = .pointer + 1;
;  3987          4047  3     symlen = . (.pointer)<0,8>;          ! get name string length
;  3988          4048  3     pointer = .pointer + 1;              ! point to string
;  3989          4049  3     if . (.prvadr)<.symval,1>
;  3990          4050  3     AND CH$NEQ(.SYMLEN, .POINTER, 7, UPLIT('UPGRADE'))          !**
;  3991          4051  3     AND CH$NEQ(.SYMLEN, .POINTER, 9, UPLIT('DOWNGRADE'))        !**
;  3992          4052  3     AND CH$NEQ(.SYMLEN, .POINTER, 6, UPLIT('TMPJNL'))           !**
;  3993          4053  3     AND CH$NEQ(.SYMLEN, .POINTER, 6, UPLIT('PRMJNL'))           !**
;  3994          4054  3     then
;  3995          4055  4         begin
;  3996          4056  4
;  3997          4057  4 !
;  3998          4058  4 ! Bit is set.  See if there's room in the buffer and insert it if so,
;  3999          4059  4 ! else output the buffer and start from scratch.
;  4000          4060  4 !
;  4001          4061  4
;  4002          4062  4         if .rabptr[rab$w_rsz] + .symlen geq 64
;  4003          4063  4         then
;  4004          4064  5             begin
;  4005          4065  5             $put (rab = .rabptr);
;  4006          4066  5             prvcnt = 0;
;  4007          4067  5             rabptr[rab$w_rsz] = 1;
;  4008          4068  4             end;
;  4009          4069  4
;  4010          4070  4 !
;  4011          4071  4 ! Insert a blank and append symbol name.
;  4012          4072  4 !
;  4013          4073  4
;  4014          4074  4         disbuf[.rabptr[rab$w_rsz]] = %char (' ');
;  4015          4075  4         rabptr[rab$w_rsz] = .rabptr[rab$w_rsz] + 1;
;  4016          4076  4         ch$move (.symlen, .pointer, disbuf[.rabptr[rab$w_rsz]]);
;  4017          4077  4         rabptr[rab$w_rsz] = .rabptr[rab$w_rsz] + .symlen;
;  4018          4078  4         prvcnt = .prvcnt + 1;            ! one more name in buffer
;  4019          4079  3         end;
;  4020          4080  3
;  4021          4081  3     pointer = .pointer + .symlen;        ! update table pointer over name
;  4022          4082  2     end;
;  4023          4083  2
;  4024          4084  2 !
;  4025          4085  2 ! Table used up.  If anything is in the buffer, print it.
;  4026          4086  2 !
;  4027          4087  2
;  4028          4088  2 if .prvcnt gtr 0
;  4029          4089  2 then $put (rab = .rabptr);
;  4030          4090  2
;  4031          4091  1 end;
```

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

                                00  45  44  41  52  47  50  55  0086C P.AFF:  .ASCII  \UPGRADE\<0>                                    ;
                00  00  00  45  44  41  52  47  4E  57  4F  44  00874 P.AFG:  .ASCII  \DOWNGRADE\<0><0><0>                           ;
                                00  00  4C  4E  4A  50  4D  54  00880 P.AFH:  .ASCII  \TMPJNL\<0><0>                               ;
                                00  00  4C  4E  4A  4D  52  50  00888 P.AFI:  .ASCII  \PRMJNL\<0><0>                               ;


                                                                        .PSECT  $CODE$,NOWRT,2

                                                                OFFC 00000 PRINT_PRIV:
                                                                           .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11             ; 3982
                        00000000'  00               20  D0 00002           MOVL    #32, DISBUF                                     ; 4029
                                                    59  D4 00009           CLRL    PRVCNT                                          ; 4030
                                   50 00000000'     00  D0 0000B           MOVL    RABPTR, R0                                      ; 4031
                                22 A0               01  B0 00012           MOVW    #1, 34(R0)                                      ;
                                   57 00000000G     00  9E 00016           MOVAB   PRV$AB_NAMES, POINTER                           ; 4032
                                   5B               67  9A 0001D 1$:        MOVZBL  (POINTER), SYMMIN                               ; 4034
                                                    03  12 00020           BNEQ    2$                                              ;
                                                0098  31 00022           BRW     6$                                              ;
                                                    57  D6 00025 2$:        INCL    POINTER                                         ; 4044
                                   5A               87  9A 00027           MOVZBL  (POINTER)+, SYMVAL                              ; 4045
                                   58               87  9A 0002A           MOVZBL  (POINTER)+, SYMLEN                              ; 4047
                        03      04 BC               5A  E0 0002D           BBS     SYMVAL, @PRVADR, 3$                             ; 4049
                                                0082  31 00032           BRW     5$                                              ;
                07              00 67               58  2D 00035 3$:        CMPC5   SYMLEN, (POINTER), #0, #7, P.AFF                ; 4050
                                      00000000'     00  0003A                                                                      ;
                                                    76  13 0003F           BEQL    5$                                              ;
                09              00 67               58  2D 00041           CMPC5   SYMLEN, (POINTER), #0, #9, P.AFG                ; 4051
                                      00000000'     00  00046                                                                      ;
                                                    6A  13 0004B           BEQL    5$                                              ;
                06              00 67               58  2D 0004D           CMPC5   SYMLEN, (POINTER), #0, #6, P.AFH                ; 4052
                                      00000000'     00  00052                                                                      ;
                                                    5E  13 00057           BEQL    5$                                              ;
                06              00 67               58  2D 00059           CMPC5   SYMLEN, (POINTER), #0, #6, P.AFI                ; 4053
                                      00000000'     00  0005E                                                                      ;
                                                    52  13 00063           BEQL    5$                                              ;
                                   50 00000000'     00  D0 00065           MOVL    RABPTR, R0                                      ; 4062
                                   51       22      A0  3C 0006C           MOVZWL  34(R0), R1                                      ;
                                   51               58  C0 00C70           ADDL2   SYMLEN, R1                                      ;
                                   3F               51  D1 00073           CMPL    R1, #63                                         ;
                                                    16  15 00076           BLEQ    4$                                              ;
                                                    50  DD 00078           PUSHL   R0                                              ; 4065
                        00000000G  00               01  FB 0007A           CALLS   #1, SYS$PUT                                     ;
                                                    59  D4 00081           CLRL    PRVCNT                                          ; 4066
                                   50 00000000'     00  D0 00083           MOVL    RABPTR, R0                                      ; 4067
                                22 A0               01  B0 0008A           MOVW    #1, 34(R0)                                      ;
                                   50 00000000'     00  D0 0008E 4$:        MOVL    RABPTR, R0                                      ; 4074
                                   56       22      A0  9E 00095           MOVAB   34(R0), R6                                      ;
                                   50               66  3C 00099           MOVZWL  (R6), R0                                        ;
                        00000000'0040               20  90 0009C           MOVB    #32, DISBUF[R0]                                 ;
                                                    66  B6 000A4           INCW    (R6)                                            ; 4075
                                   50               66  3C 000A6           MOVZWL  (R6), R0                                        ; 4076
                00000000'004C      67               58  28 000A9           MOVC3   SYMLEN, (POINTER), DISBUF[R0]                   ;
                                   66               58  A0 000B2           ADDW2   SYMLEN, (R6)                                    ; 4077
                                                    59  D6 000B5           INCL    PRVCNT                                          ; 4078
```

```
                                57            58  C0 000B7 5$:     ADDL2   SYMLEN, POINTER                ; 4081
                              FF60            31  000BA           BRW     1$                              ; 4034
                                59            D5  000BD 6$:      TSTL    PRVCNT                          ; 4088
                                0D            15  000BF           BLEQ    7$                              ;
                     00000000'  00            DD  000C1           PUSHL   RABPTR                          ; 4089
          00000000G  00                       01  FB 000C7       CALLS   #1, SYS$PUT                     ;
                                                  04  000CE 7$:      RET                                 ; 4091

; Routine Size:  207 bytes,    Routine Base:  $CODE$ + 21F6
```

```
;  4033     4092  1 %sbttl 'build_ini_recs - build system & default records'
;  4034     4093  1 routine build_ini_recs : novalue =
;  4035     4094  2 begin
;  4036     4095  2
;  4037     4096  2 !++
;  4038     4097  2 !
;  4039     4098  2 ! FUNCTIONAL DESCRIPTION:
;  4040     4099  2 !
;  4041     4100  2 !       Build the initial records for the creation of a new UAF file.
;  4042     4101  2 !       The user default record is built in the DEFAULT_RECORD
;  4043     4102  2 !       buffer and the system manager record is built in RECBUF.
;  4044     4103  2 !
;  4045     4104  2 ! INPUTS:
;  4046     4105  2 !
;  4047     4106  2 !       none
;  4048     4107  2 !
;  4049     4108  2 ! IMPLICIT INPUTS:
;  4050     4109  2 !
;  4051     4110  2 !       none
;  4052     4111  2 !
;  4053     4112  2 ! OUTPUTS:
;  4054     4113  2 !
;  4055     4114  2 !       none
;  4056     4115  2 !
;  4057     4116  2 ! IMPLICIT OUTPUTS:
;  4058     4117  2 !
;  4059     4118  2 !       default record is built in DEFAULT_RECORD
;  4060     4119  2 !       system record is built in RECBUF
;  4061     4120  2 !
;  4062     4121  2 ! ROUTINE VALUE:
;  4063     4122  2 !
;  4064     4123  2 !       none
;  4065     4124  2 !
;  4066     4125  2 ! SIDE EFFECTS:
;  4067     4126  2 !
;  4068     4127  2 !       none
;  4069     4128  2 !--
;  4070     4129  2
;  4071     4130  2 local
;  4072     4131  2         user_desc       : $bblock [8];
;  4073     4132  2
;  4074     4133  2 ch$fill (0, uaf$c_fixed, default_record);
;  4075     4134  2 default_record[uaf$b_version] = uaf$c_version1;
;  4076     4135  2 default_record[uaf$b_rtype] = uaf$c_user_id;
;  4077     4136  2 !
;  4078     4137  2 ! username is blank filled
;  4079     4138  2 !
;  4080     4139  2
;  4081     4140  2 ch$copy (.defuser<0,8>, defuser+1, %char (' '),
;  4082     4141  2           uaf$s_username, default_record[uaf$t_username]);
;  4083     4142  2
;  4084     4143  2 !
;  4085     4144  2 ! account name is blank filled
;  4086     4145  2 !
;  4087     4146  2
;  4088     4147  2 ch$copy (.defact<0,8>, defact+1, %char (' '),
;  4089     4148  2           uaf$s_account, default_record[uaf$t_account]);
```

```
;  4090          4149  2
;  4091          4150  2 !
;  4092          4151  2 ! quadword privilege mask
;  4093          4152  2 !
;  4094          4153  2
;  4095          4154  2 ch$move (8, defpriv, default_record[uaf$q_priv]);
;  4096          4155  2 ch$move (8, defpriv, default_record[uaf$q_def_priv]);
;  4097          4156  2
;  4098          4157  2 !
;  4099          4158  2 ! directory name is counted string
;  4100          4159  2 !
;  4101          4160  2
;  4102          4161  2 ch$copy (.defdir<0,8> + 1, defdir, %char (' '),
;  4103          4162  2           uaf$s_defdir, default_record[uaf$t_defdir]);
;  4104          4163  2
;  4105          4164  2 !
;  4106          4165  2 ! device name is counted string
;  4107          4166  2 !
;  4108          4167  2
;  4109          4168  2 ch$copy (.defdev<0,8> + 1, defdev, %char (' '),
;  4110          4169  2           uaf$s_defdev, default_record[uaf$t_defdev]);
;  4111          4170  2
;  4112          4171  2 !
;  4113          4172  2 ! CLI name is counted string
;  4114          4173  2 !
;  4115          4174  2
;  4116          4175  2 ch$copy (.defcli<0,8> + 1, defcli, %char (' '),
;  4117          4176  2           uaf$s_defcli, default_record[uaf$t_defcli]);
;  4118          4177  2
;  4119          4178  2 !
;  4120          4179  2 ! owner name is counted string
;  4121          4180  2 !
;  4122          4181  2
;  4123          4182  2 ch$copy (.defowner<0,8> + 1, defowner, %char (' '),
;  4124          4183  2           uaf$s_owner, default_record[uaf$t_owner]);
;  4125          4184  2
;  4126          4185  2 !
;  4127          4186  2 ! login command file name is counted string
;  4128          4187  2 !
;  4129          4188  2
;  4130          4189  2 ch$copy (.deflgicmd<0,8> + 1, deflgicmd, %char (' '),
;  4131          4190  2           uaf$s_lgicmd, default_record[uaf$t_lgicmd]);
;  4132          4191  2
;  4133          4192  2 !
;  4134          4193  2 ! fill in default CLI tables
;  4135          4194  2 !
;  4136          4195  2 ch$copy (.defclitabl<0,8> + 1, defclitabl, %char (' '),
;  4137          4196  2           uaf$s_clitables, default_record[uaf$t_clitables]);
;  4138          4197  2
;  4139          4198  2 ch$move (8, defpwdlife, default_record[uaf$q_pwd_lifetime]);
;  4140          4199  2 default_record[uaf$b_pwd_length] = defpwdlength;
;  4141          4200  2 default_record[uaf$w_grp] = defgrp;
;  4142          4201  2 default_record[uaf$w_mem] = defmem;
;  4143          4202  2 default_record[uaf$w_biolm] = defbiolm;
;  4144          4203  2 default_record[uaf$l_bytlm] = defbytlm;
;  4145          4204  2 default_record[uaf$w_diolm] = defdiolm;
;  4146          4205  2 default_record[uaf$w_fillm] = deffillm;
```

```
;  4147          4206  2 default_record[uaf$l_flags] = defflags;
;  4148          4207  2 default_record[uaf$w_tqcnt] = deftqcnt;
;  4149          4208  2 default_record[uaf$w_prccnt] = defprccnt;
;  4150          4209  2 default_record[uaf$l_wsquota] = defwsquota;
;  4151          4210  2 default_record[uaf$l_wsextent] = defwsextent;
;  4152          4211  2 default_record[uaf$l_dfwscnt] = defdfwscnt;
;  4153          4212  2 default_record[uaf$l_cputim] = defcputim;
;  4154          4213  2 default_record[uaf$w_astlm] = defastlm;
;  4155          4214  2 default_record[uaf$l_pgflquota] = defpgflquota;
;  4156          4215  2 default_record[uaf$w_enqlm] = defenqlm;
;  4157          4216  2 default_record[uaf$l_pbytlm] = defpbytlm;
;  4158          4217  2 default_record[uaf$w_shrfillm] = defshrfillm;
;  4159          4218  2 default_record[uaf$b_pri] = defpri;
;  4160          4219  2 default_record[uaf$b_quepri] = defquepri;
;  4161          4220  2 default_record[uaf$w_maxjobs] = defmaxjobs;
;  4162          4221  2 default_record[uaf$w_maxdetach] = defmaxdetach;
;  4163          4222  2 default_record[uaf$l_jtquota] = defjtquota;
;  4164          4223  2 default_record[uaf$w_maxacctjobs] = defmaxacctjobs;
;  4165          4224  2 default_record[uaf$b_primedays] = defprimedays;
;  4166          4225  2 default_record[uaf$b_network_access_p] = defhours;
;  4167          4226  2 default_record[uaf$b_network_access_s] = defhours;
;  4168          4227  2 default_record[uaf$b_batch_access_p] = defhours;
;  4169          4228  2 default_record[uaf$b_batch_access_s] = defhours;
;  4170          4229  2 default_record[uaf$b_local_access_p] = defhours;
;  4171          4230  2 default_record[uaf$b_local_access_s] = defhours;
;  4172          4231  2 default_record[uaf$b_dialup_access_p] = defhours;
;  4173          4232  2 default_record[uaf$b_dialup_access_s] = defhours;
;  4174          4233  2 default_record[uaf$b_remote_access_p] = defhours;
;  4175          4234  2 default_record[uaf$b_remote_access_s] = defhours;
;  4176          4235  2
;  4177          4236  2 ch$fill (0, uaf$c_fixed, recbuf);
;  4178          4237  2 recbuf[uaf$b_version] = uaf$c_version1;
;  4179          4238  2 recbuf[uaf$b_rtype] = uaf$c_user_id;
;  4180          4239  2 ch$copy (.sysuser<0,8>, sysuser+1, %char (' '),
;  4181          4240  2           uaf$s_username, recbuf[uaf$t_username]);
;  4182          4241  2 ch$copy (.sysact<0,8>, sysact+1, %char (' '),
;  4183          4242  2           uaf$s_account, recbuf[uaf$t_account]);
;  4184          4243  2 ch$move (8, syspriv, recbuf[uaf$q_priv]);
;  4185          4244  2 ch$move (8, syspriv, recbuf[uaf$q_def_priv]);
;  4186          4245  2 ch$copy (.sysdir<0,8> + 1, sysdir, %char (' '),
;  4187          4246  2           uaf$s_defdir, recbuf[uaf$t_defdir]);
;  4188          4247  2 ch$copy (.sysdev<0,8> + 1, sysdev, %char (' '),
;  4189          4248  2           uaf$s_defdev, recbuf[uaf$t_defdev]);
;  4190          4249  2 ch$copy (.syscli<0,8> + 1, syscli, %char (' '),
;  4191          4250  2           uaf$s_defcli, recbuf[uaf$t_defcli]);
;  4192          4251  2 ch$copy (.sysowner<0,8> + 1, sysowner, %char (' '),
;  4193          4252  2           uaf$s_owner, recbuf[uaf$t_owner]);
;  4194          4253  2 ch$copy (.syslgicmd<0,8> + 1, deflgicmd, %char (' '),
;  4195          4254  2           uaf$s_lgicmd, recbuf[uaf$t_lgicmd]);
;  4196          4255  2 ch$copy (.sysclitabl<0,8> + 1, sysclitabl, %char (' '),
;  4197          4256  2           uaf$s_clitables, recbuf[uaf$t_clitables]);
;  4198          4257  2
;  4199          4258  2 pwddsc[dsc$w_length] = .syspass<0,8>;
;  4200          4259  2 pwddsc[dsc$a_pointer] = syspass+1;
;  4201          4260  2 user_desc[dsc$w_length] = .sysuser<0,8>;
;  4202          4261  2 user_desc[dsc$a_pointer] = sysuser+1;
;  4203          4262  2 $gettim (timadr = time_buf);                    ! Obtain a 16 bit salt
```

```
;  4204          4263  2 recbuf[uaf$w_salt] = .time_buf<3*8,16>;
;  4205          4264  2 recbuf[uaf$b_encrypt] = encrypt;
;  4206          4265  2 lgi$hpwd (rec_encrypt_dsc, pwddsc, .recbuf[uaf$b_encrypt],
;  4207          4266  2             .recbuf[uaf$w_salt], user_desc);
;  4208          4267  2
;  4209          4268  2 ch$move (8, syspwdlife, recbuf[uaf$q_pwd_lifetime]);
;  4210          4269  2 recbuf[uaf$b_pwd_length] = syspwdlength;
;  4211          4270  2 recbuf[uaf$w_grp] = sysgrp;
;  4212          4271  2 recbuf[uaf$w_mem] = sysmem;
;  4213          4272  2 recbuf[uaf$w_biolm] = sysbiolm;
;  4214          4273  2 recbuf[uaf$l_bytlm] = sysbytlm;
;  4215          4274  2 recbuf[uaf$w_diolm] = sysdiolm;
;  4216          4275  2 recbuf[uaf$w_fillm] = sysfillm;
;  4217          4276  2 recbuf[uaf$l_flags] = sysflags;
;  4218          4277  2 recbuf[uaf$w_tqcnt] = systqcnt;
;  4219          4278  2 recbuf[uaf$w_prccnt] = sysprccnt;
;  4220          4279  2 recbuf[uaf$l_wsquota] = syswsquota;
;  4221          4280  2 recbuf[uaf$l_wsextent] = syswsextent;
;  4222          4281  2 recbuf[uaf$l_dfwscnt] = sysdfwscnt;
;  4223          4282  2 recbuf[uaf$l_cputim] = syscputim;
;  4224          4283  2 recbuf[uaf$w_astlm] = sysastlm;
;  4225          4284  2 recbuf[uaf$l_pgflquota] = syspgflquota;
;  4226          4285  2 recbuf[uaf$w_enqlm] = sysenqlm;
;  4227          4286  2 recbuf[uaf$l_pbytlm] = syspbytlm;
;  4228          4287  2 recbuf[uaf$w_shrfillm] = sysshrfillm;
;  4229          4288  2 recbuf[uaf$b_pri] = syspri;
;  4230          4289  2 default_record[uaf$b_quepri] = sysquepri;
;  4231          4290  2 recbuf[uaf$w_maxdetach] = sysmaxdetach;
;  4232          4291  2 recbuf[uaf$l_jtquota] = sysjtquota;
;  4233          4292  2 recbuf[uaf$w_maxjobs] = sysmaxjobs;
;  4234          4293  2 recbuf[uaf$w_maxdetach] = sysmaxdetach;
;  4235          4294  2 recbuf[uaf$w_maxacctjobs] = sysmaxacctjobs;
;  4236          4295  2 recbuf[uaf$b_primedays] = sysprimedays;
;  4237          4296  2 recbuf[uaf$b_network_access_p] = defhours;
;  4238          4297  2 recbuf[uaf$b_network_access_s] = defhours;
;  4239          4298  2 recbuf[uaf$b_batch_access_p] = defhours;
;  4240          4299  2 recbuf[uaf$b_batch_access_s] = defhours;
;  4241          4300  2 recbuf[uaf$b_local_access_p] = defhours;
;  4242          4301  2 recbuf[uaf$b_local_access_s] = defhours;
;  4243          4302  2 recbuf[uaf$b_dialup_access_p] = defhours;
;  4244          4303  2 recbuf[uaf$b_dialup_access_s] = defhours;
;  4245          4304  2 recbuf[uaf$b_remote_access_p] = defhours;
;  4246          4305  2 recbuf[uaf$b_remote_access_s] = defhours;
;  4247          4306  1 end;


                                        03FC 00000 BUILD_INI_RECS:
                                                   .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9                    ; 4093
                        59 00000000'    00  9E 00002       MOVAB   DEFACT+1, R9                            ;
                        58 00000000'    00  9E 00009       MOVAB   DEFAULT_RECORD, R8                      ;
                        57 00000000'    00  9E 00010       MOVAB   RECBUF, R7                              ;
                        5E              08  C2 00017       SUBL2   #8, SP                                  ;
    0284    8F          00  6E          00  2C 0001A       MOVC5   #0, (SP), #0, #644, DEFAULT_RECORD      ; 4133
                                        68     00021                                                       ;
```

```
                                                68  0101  8F  B0 00022      MOVW    #257, DEFAULT_RECORD                                    ; 4135
                                                50    E8  A9  9A 00027      MOVZBL  DEFUSER, R0                                             ; 4140
          20            20        E9  A9                  50  2C 0002B      MOVC5   R0, DEFUSER+1, #32, #32, DEFAULT_RECORD+4               ; 4141
                                                04          A8     00031
                                                50    FF  A9  9A 00033      MOVZBL  DEFACT, R0                                              ; 4147
          20            20            69                  50  2C 00037      MOVC5   R0, DEFACT+1, #32, #32, DEFAULT_RECORD+52               ; 4148
                                                34          A8     0003C
                019C    C8        10  A9                  08  28 0003E      MOVC3   #8, DEFPRIV, DEFAULT_RECORD+412                         ; 4154
                01A4    C8        10  A9                  08  28 00045      MOVC3   #8, DEFPRIV, DEFAULT_RECORD+420                         ; 4155
                                                50    06  A9  9A 0004C      MOVZBL  DEFDIR, R0                                              ; 4161
                                                            50  D6 00050      INCL    R0
    0040    8F            20        06  A9                  50  2C 00052      MOVC5   R0, DEFDIR, #32, #64, DEFAULT_RECORD+148                ; 4162
                                                0094        C8     0005A
                                                50    0D  A9  9A 0005D      MOVZBL  DEFDEV, R0                                              ; 4168
                                                            50  D6 00061      INCL    R0
          20            20        0D  A9                  50  2C 00063      MOVC5   R0, DEFDEV, #32, #32, DEFAULT_RECORD+116                ; 4169
                                                74          A8     00069
                                                50          69  9A 0006B      MOVZBL  DEFCLI, R0                                              ; 4175
                                                            50  D6 0006E      INCL    R0
          20            20            69                  50  2C 00070      MOVC5   R0, DEFCLI, #32, #32, DEFAULT_RECORD+276                ; 4176
                                                0114        C8     00075
                                                50    04  A9  9A 00078      MOVZBL  DEFOWNER, R0                                            ; 4182
                                                            50  D6 0007C      INCL    R0
          20            20        04  A9                  50  2C 0007E      MOVC5   R0, DEFOWNER, #32, #32, DEFAULT_RECORD+84               ; 4183
                                                54          A8     00084
                                                50    05  A9  9A 00086      MOVZBL  DEFLGICMD, R0                                           ; 4189
                                                            50  D6 0008A      INCL    R0
    0040    8F            20        05  A9                  50  2C 0008C      MOVC5   R0, DEFLGICMD, #32, #64, DEFAULT_RECORD+212             ; 4190
                                                00D4        C8     00094
                                                50    F5  A9  9A 00097      MOVZBL  DEFCLITABL, R0                                          ; 4195
                                                            50  D6 0009B      INCL    R0
          20            20        F5  A9                  50  2C 0009D      MOVC5   R0, DEFCLITABL, #32, #32, -                             ; 4196
                                                0134        C8     000A3              DEFAULT_RECORD+308
                0174    C8        18  A9                  08  28 000A6      MOVC3   #8, DEFPWDLIFE, DEFAULT_RECORD+372                      ; 4198
                                    016A    C8              06  90 000AD      MOVB    #6, DEFAULT_RECORD+362                                  ; 4199
                                      24    A8  00800080    8F  D0 000B2      MOVL    #8388736, DEFAULT_RECORD+36                             ; 4201
                                    0230    C8      1000    8F  3C 000BA      MOVZWL  #4096, DEFAULT_RECORD+560                               ; 4203
                                    020E    C8  00060006    8F  D0 000C1      MOVL    #393222, DEFAULT_RECORD+526                             ; 4202
                                                01D4        C8  D4 000CA      CLRL    DEFAULT_RECORD+468                                      ; 4206
                                    020C    C8              02  B0 000CE      MOVW    #2, DEFAULT_RECORD+524                                  ; 4208
                                    021C    C8          C8  8F  9A 000D3      MOVZBL  #200, DEFAULT_RECORD+540                                ; 4209
                                    0224    C8        01F4  8F  3C 000D9      MOVZWL  #500, DEFAULT_RECORD+548                                ; 4210
                                    0220    C8          96  8F  9A 000E0      MOVZBL  #150, DEFAULT_RECORD+544                                ; 4211
                                                022C        C8  D4 000E6      CLRL    DEFAULT_RECORD+556                                      ; 4212
                                    0212    C8  000A000A    8F  D0 000EA      MOVL    #655370, DEFAULT_RECORD+530                             ; 4207
                                    0228    C8        2710  8F  3C 000F3      MOVZWL  #10000, DEFAULT_RECORD+552                              ; 4214
                                    0216    C8  0014000A    8F  D0 000FA      MOVL    #1310730, DEFAULT_RECORD+534                            ; 4215
                                                0234        C8  D4 00103      CLRL    DEFAULT_RECORD+564                                      ; 4216
                                                021A        C8  B4 00107      CLRW    DEFAULT_RECORD+538                                      ; 4217
                                    0204    C8        0404  8F  3C 0010B      MOVZWL  #1028, DEFAULT_RECORD+516                               ; 4218
                                    0238    C8        0400  8F  3C 00112      MOVZWL  #1024, DEFAULT_RECORD+568                               ; 4222
                                                0208        C8  D4 00119      CLRL    DEFAULT_RECORD+520                                      ; 4223
                                    0202    C8          60  8F  90 0011D      MOVB    #96, DEFAULT_RECORD+514                                 ; 4224
    01D8    C8            18              00              00  F0 00123      INSV    #0, #0, #24, DEFAULT_RECORD+472                         ; 4225
    01DB    C8            18              00              00  F0 0012A      INSV    #0, #0, #24, DEFAULT_RECORD+475                         ; 4226
    01DE    C8            18              00              00  F0 00131      INSV    #0, #0, #24, DEFAULT_RECORD+478                         ; 4227
    01E1    C8            18              00              00  F0 00138      INSV    #0, #0, #24, DEFAULT_RECORD+481                         ; 4228
```

```
 01E4  C8        18          00           00  F0 0013F     INSV    #0, #0, #24, DEFAULT_RECORD+484        ; 4229
 01E7  C8        18          00           00  F0 00146     INSV    #0, #0, #24, DEFAULT_RECORD+487        ; 4230
 01EA  C8        18          00           00  F0 0014D     INSV    #0, #0, #24, DEFAULT_RECORD+490        ; 4231
 01ED  C8        18          00           00  F0 00154     INSV    #0, #0, #24, DEFAULT_RECORD+493        ; 4232
 01F0  C8        18          00           00  F0 0015B     INSV    #0, #0, #24, DEFAULT_RECORD+496        ; 4233
 01F3  C8        18          00           00  F0 00162     INSV    #0, #0, #24, DEFAULT_RECORD+499        ; 4234
 0284  8F        00          6E           00  2C 00169     MOVC5   #0, (SP), #0, #644, RECBUF             ; 4236
                                          67     00170                                                  ;
                             67     0101  8F  B0 00171     MOVW    #257, RECBUF                           ; 4238
                             56       20  A9  9A 00176     MOVZBL  SYSUSER, R6                            ; 4239
       20        20    21    A9           56  2C 0017A     MOVC5   R6, SYSUSER+1, #32, #32, RECBUF+4      ; 4240
                             04           A7     00180                                                  ;
                             50       39  A9  9A 00182     MOVZBL  SYSACT, R0                             ; 4241
       20        20    3A    A9           50  2C 00186     MOVC5   R0, SYSACT+1, #32, #32, RECBUF+52      ; 4242
                                      34  A7     0018C                                                  ;
            019C  C7   60    A9           08  28 0018E     MOVC3   #8, SYSPRIV, RECBUF+412                ; 4243
            01A4  C7   60    A9           08  28 00195     MOVC3   #8, SYSPRIV, RECBUF+420                ; 4244
                             50       54  A9  9A 0019C     MOVZBL  SYSDIR, R0                             ; 4245
                                          50  D6 001A0     INCL    R0                                     ;
 0040  8F        20    54    A9           50  2C 001A2     MOVC5   R0, SYSDIR, #32, #64, RECBUF+148       ; 4246
                                    0094  C7     001AA                                                  ;
                             50       5D  A9  9A 001AD     MOVZBL  SYSDEV, R0                             ; 4247
                                          50  D6 001B1     INCL    R0                                     ;
       20        20    5D    A9           50  2C 001B3     MOVC5   R0, SYSDEV, #32, #32, RECBUF+116       ; 4248
                                      74  A7     001B9                                                  ;
                             50       40  A9  9A 001BB     MOVZBL  SYSCLI, R0                             ; 4249
                                          50  D6 001BF     INCL    R0                                     ;
       20        20    40    A9           50  2C 001C1     MOVC5   R0, SYSCLI, #32, #32, RECBUF+276       ; 4250
                                    0114  C7     001C7                                                  ;
                             50       44  A9  9A 001CA     MOVZBL  SYSOWNER, R0                           ; 4251
                                          50  D6 001CE     INCL    R0                                     ;
       20        20    44    A9           50  2C 001D0     MOVC5   R0, SYSOWNER, #32, #32, RECBUF+84      ; 4252
                                      54  A7     001D6                                                  ;
                             50       53  A9  9A 001D8     MOVZBL  SYSLGICMD, R0                          ; 4253
                                          50  D6 001DC     INCL    R0                                     ;
 0040  8F        20    05    A9           50  2C 001DE     MOVC5   R0, DEFLGICMD, #32, #64, RECBUF+212    ; 4254
                                    00D4  C7     001E6                                                  ;
                             50       2F  A9  9A 001E9     MOVZBL  SYSCLITABL, R0                         ; 4255
                                          50  D6 001ED     INCL    R0                                     ;
       20        20    2F    A9           50  2C 001EF     MOVC5   R0, SYSCLITABL, #32, #32, RECBUF+308   ; 4256
                                    0134  C7     001F5                                                  ;
                     0584    C8       27  A9  9B 001F8     MOVZBW  SYSPASS, PWDDSC                        ; 4258
                     0588    C8       28  A9  9E 001FE     MOVAB   SYSPASS+1, PWDDSC+4                    ; 4259
                             6E           56  B0 00204     MOVW    R6, USER_DESC                          ; 4260
                       04    AE       21  A9  9E 00207     MOVAB   SYSUSER+T, USER_DESC+4                 ; 4261
                                    0670  C7  9F 0020C     PUSHAB  TIME_BUF                               ; 4262
                00000000G    00           01  FB 00210     CALLS   #1, SYS$GETTIM                         ;
                     0166    C7     0673  C7  B0 00217     MOVW    TIME_BUF+3, RECBUF+358                 ; 4263
                     0168    C7           02  90 0021E     MOVB    #2, RECBUF+360                         ; 4264
                                          5E  DD 00223     PUSHL   SP                                     ; 4265
                             7E     0166  C7  3C 00225     MOVZWL  RECBUF+358, -(SP)                      ; 4266
                             7E     0168  C7  9A 0022A     MOVZBL  RECBUF+360, -(SP)                      ; 4265
                                    0584  C8  9F 0022F     PUSHAB  PWDDSC                                 ;
                                    00B4  C9  9F 00233     PUSHAB  REC_ENCRYPT_DSC                        ;
                00000000G    00           05  FB 00237     CALLS   #5, LGI$HPWD                           ;
            0174  C7   68    A9           08  28 0023E     MOVC3   #8, SYSPWDLIFE, RECBUF+372             ; 4268
                     016A    C7           08  90 00245     MOVB    #8, RECBUF+362                         ; 4269
```

```
                                  24    A7 00010004    8F   D0 0024A        MOVL    #65540, RECBUF+36                       ; 4271
                                0230    C7     5000    8F   3C 00252        MOVZWL  #20480, RECBUF+560                      ; 4273
                                020E    C7 000C000C    8F   D0 00259        MOVL    #786444, RECBUF+526                     ; 4272
                                            01D4    C7   D4 00262        CLRL    RECBUF+468                              ; 4276
                                0212    C7 00140014    8F   D0 00266        MOVL    #1310740, RECBUF+530                    ; 4277
                                020A    C7 000A0000    8F   D0 0026F        MOVL    #655360, RECBUF+522                     ; 4290
                                021C    C7     015E    8F   3C 00278        MOVZWL  #350, RECBUF+540                        ; 4279
                                0224    C7     0400    8F   3C 0027F        MOVZWL  #1024, RECBUF+548                       ; 4280
                                0220    C7       96    8F   9A 00286        MOVZBL  #150, RECBUF+544                        ; 4281
                                            022C    C7   D4 0028C        CLRL    RECBUF+556                              ; 4282
                                0228    C7     2710    8F   3C 00290        MOVZWL  #10000, RECBUF+552                      ; 4284
                                0216    C7 00140014    8F   D0 00297        MOVL    #1310740, RECBUF+534                    ; 4285
                                            0234    C7   D4 002A0        CLRL    RECBUF+564                              ; 4286
                                            021A    C7   B4 002A4        CLRW    RECBUF+538                              ; 4287
                                0204    C7             04   90 002A8        MOVB    #4, RECBUF+516                          ; 4288
                                0205    C8             04   90 002AD        MOVB    #4, DEFAULT_RECORD+517                  ; 4289
                                0238    C7     0400    8F   3C 002B2        MOVZWL  #1024, RECBUF+568                       ; 4291
                                            020A    C7   B4 002B9        CLRW    RECBUF+522                              ; 4293
                                            0206    C7   D4 002BD        CLRL    RECBUF+518                              ; 4292
                                0202    C7       60    8F   90 002C1        MOVB    #96, RECBUF+514                         ; 4295
    01D8    C7          18              00              00   F0 002C7        INSV    #0, #0, #24, RECBUF+472                 ; 4296
    01DB    C7          18              00              00   F0 002CE        INSV    #0, #0, #24, RECBUF+475                 ; 4297
    01DE    C7          18              00              00   F0 002D5        INSV    #0, #0, #24, RECBUF+478                 ; 4298
    01E1    C7          18              00              00   F0 002DC        INSV    #0, #0, #24, RECBUF+481                 ; 4299
    01E4    C7          18              00              00   F0 002E3        INSV    #0, #0, #24, RECBUF+484                 ; 4300
    01E7    C7          18              00              00   F0 002EA        INSV    #0, #0, #24, RECBUF+487                 ; 4301
    01EA    C7          18              00              00   F0 002F1        INSV    #0, #0, #24, RECBUF+490                 ; 4302
    01ED    C7          18              00              00   F0 002F8        INSV    #0, #0, #24, RECBUF+493                 ; 4303
    01F0    C7          18              00              00   F0 002FF        INSV    #0, #0, #24, RECBUF+496                 ; 4304
    01F3    C7          18              00              00   F0 00306        INSV    #0, #0, #24, RECBUF+499                 ; 4305
                                                            04   0030D        RET                                             ; 4306

; Routine Size:  782 bytes,    Routine Base:  $CODE$ + 22C5
```

```
;  4249     4307  1 %sbttl 'get_user_record - get username and lookup record'
;  4250     4308  1 routine get_user_record (lock_record, permanent_ok) =
;  4251     4309  2 begin
;  4252     4310  2
;  4253     4311  2 !++
;  4254     4312  2 !
;  4255     4313  2 ! FUNCTIONAL DESCRIPTION:
;  4256     4314  2 !
;  4257     4315  2 !       This routine pulls the next token out of the command
;  4258     4316  2 !       buffer, assuming it is the username, and looks up the
;  4259     4317  2 !       UAF record for that name.  If the record is found,
;  4260     4318  2 !       it is loaded into RECBUF (by routine LOCATE_USER) .
;  4261     4319  2 !
;  4262     4320  2 ! INPUTS:
;  4263     4321  2 !
;  4264     4322  2 !       LOCK_RECORD - specifies that the GET shall lock the record
;  4265     4323  2 !       PERMANENT_OK - specifies that the DEFAULT and SYSTEM records are allowed
;  4266     4324  2 !
;  4267     4325  2 ! IMPLICIT INPUTS:
;  4268     4326  2 !
;  4269     4327  2 !       TOKENPTR - address of delimiter following last token processed,
;  4270     4328  2 !                  which was the command name.
;  4271     4329  2 !       TOKENLEN - global variable to contain length of current token
;  4272     4330  2 !
;  4273     4331  2 ! OUTPUTS:
;  4274     4332  2 !
;  4275     4333  2 !       none
;  4276     4334  2 !
;  4277     4335  2 ! IMPLICIT OUTPUTS:
;  4278     4336  2 !
;  4279     4337  2 !       none
;  4280     4338  2 !
;  4281     4339  2 ! ROUTINE VALUE:
;  4282     4340  2 !
;  4283     4341  2 !       true -> user record found
;  4284     4342  2 !       false -> user record not found
;  4285     4343  2 !
;  4286     4344  2 ! SIDE EFFECTS:
;  4287     4345  2 !
;  4288     4346  2 !       none
;  4289     4347  2 !--
;  4290     4348  2
;  4291     4349  2 builtin
;  4292     4350  2         nullparameter;
;  4293     4351  2
;  4294     4352  2 !
;  4295     4353  2 ! Is this the second phase of a RENAME?
;  4296     4354  2 !
;  4297     4355  2 if .rename_ph2
;  4298     4356  2 then
;  4299     4357  3     begin
;  4300     4358  3     if .netuaf_exists
;  4301     4359  3     then adjust_proxy (update_records);
;  4302     4360  3     ch$move (.olduserlen, oldusername, .tokenptr);
;  4303     4361  3     tokenlen = .olduserlen;
;  4304     4362  3     rename_ph2 = false;
;  4305     4363  3     end
```

```
;  4306     4364  3 !
;  4307     4365  3 ! Not the second phase of a RENAME.  Get first token, and if this
;  4308     4366  3 ! is the first phase of a RENAME (COPY phase), save token for
;  4309     4367  3 ! the second phase (REMOVE phase).
;  4310     4368  3 !
;  4311     4369  2 else
;  4312     4370  2     begin
;  4313     4371  3 !
;  4314     4372  3 ! Get token
;  4315     4373  3 !
;  4316     4374  3         if not cli$present (sd_token1)
;  4317     4375  3         or not cli$get_value (sd_token1,tokendsc)
;  4318     4376  3         or .tokenlen eql 0
;  4319     4377  3         then return LIB$SIGNAL(UAF$_NOUSERNAME);
;  4320     4378  3 !
;  4321     4379  3 ! If the third argument is present, this is the first phase of a
;  4322     4380  3 ! RENAME,  so save the token for the next call
;  4323     4381  3 !
;  4324     4382  3         if not nullparameter (3)
;  4325     4383  3         then
;  4326     4384  4             begin
;  4327     4385  4             ch$copy (.tokenlen, .tokenptr, ' ', uaf$s_username, oldusername);
;  4328     4386  4             olduserlen = .tokenlen;
;  4329     4387  4             rename_ph2 = true;
;  4330     4388  3             end;
;  4331     4389  2         end;
;  4332     4390  2
;  4333     4391  2 if not .permanent_ok
;  4334     4392  2 then
;  4335     4393  3     begin
;  4336     4394  3     if ch$eql (.defuser<0,8>, defuser+1, .tokenlen, .tokenptr, ' ')
;  4337     4395  3     then
;  4338     4396  3         if nullparameter (3)
;  4339     4397  3         then
;  4340     4398  3             return LIB$SIGNAL(UAF$_REMDEF)
;  4341     4399  3         else
;  4342     4400  3             return LIB$SIGNAL(UAF$_RENDEF);
;  4343     4401  3
;  4344     4402  3     if ch$eql (.sysuser<0,8>, sysuser+1, .tokenlen, .tokenptr, ' ')
;  4345     4403  3     then
;  4346     4404  4         begin
;  4347     4405  4         if nullparameter (3)
;  4348     4406  4         then
;  4349     4407  4             return LIB$SIGNAL(UAF$_REMSYS)
;  4350     4408  4         else
;  4351     4409  4             return LIB$SIGNAL(UAF$_RENSYS);
;  4352     4410  3         end;
;  4353     4411  2     end;
;  4354     4412  2
;  4355     4413  2 if locate_user (.tokenlen, .tokenptr, .lock_record)
;  4356     4414  2 then return true;
;  4357     4415  2
;  4358     4416  2 if .rmserr eql rms$_rnf
;  4359     4417  2 then
;  4360     4418  2     LIB$SIGNAL(UAF$_BADUSR, 2, .tokenlen, .tokenptr)
;  4361     4419  2 else
;  4362     4420  2     LIB$SIGNAL(UAF$_GETERR, 0, .rmserr);
```

```
;  4363          4421  2 false
;  4364          4422  1 end;


                                                07FC 00000 GET_USER_RECORD:
                                                           .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10                ; 4308
                                5A 00000000G    00   9E 00002  MOVAB   LIB$SIGNAL, R10                           ;
                                59 00000000'    00   9E 00009  MOVAB   SD_TOKEN1, R9                            ;
                                58 00000000'    00   9E 00010  MOVAB   RENAME_PH2, R8                          ;
                                57 00000000'    00   9E 00017  MOVAB   TOKENLEN, R7                            ;
                                1F              68   E9 0001E  BLBC    RENAME_PH2, 2$                          ; 4355
                                07        10    A7   E9 00021  BLBC    NETUAF_EXISTS, 1$                       ; 4358
                                          7E    D4   00025     CLRL    -(SP)                                   ; 4359
                          E796  CF              01   FB 00027  CALLS   #1, ADJUST_PROXY                        ;
                                56        04    A8   D0 0002C  1$:     MOVL    OLDUSERLEN, R6                  ; 4360
                                50        04    A7   D0 00030  MOVL    TOKENPTR, R0                            ;
                    60    08  A8              56   28 00034  MOVC3   R6, OLDUSERNAME, (R0)                     ;
                                67              56   B0 00039  MOVW    R6, TOKENLEN                            ; 4361
                                                68   94 0003C  CLRB    RENAME_PH2                              ; 4362
                                                45   11 0003E  BRB     5$                                      ; 4355
                                                59   DD 00040  2$:     PUSHL   R9                              ; 4374
                      00000000G 00              01   FB 00042  CALLS   #1, CLI$PRESENT                         ;
                                12              50   E9 00049  BLBC    R0, 3$                                  ;
                                                57   DD 0004C  PUSHL   R7                                      ; 4375
                                                59   DD 0004E  PUSHL   R9                                      ;
                      00000000G 00              02   FB 00050  CALLS   #2, CLI$GET_VALUE                       ;
                                04              50   E9 00057  BLBC    R0, 3$                                  ;
                                                67   B5 0005A  TSTW    TOKENLEN                                ; 4376
                                                08   12 0005C  BNEQ    4$                                      ;
                           00000000G            8F   DD 0005E  3$:     PUSHL   #UAF$_NOUSERNAME                ; 4377
                                                7B   11 00064  BRB     11$                                     ;
                                03              6C   91 00066  4$:     CMPB    (AP), #3                        ; 4382
                                                1A   1F 00069  BLSSU   5$                                      ;
                                      0C        AC   D5 0006B  TSTL    12(AP)                                  ;
                                                15   13 0006E  BEQL    5$                                      ;
                                56              67   3C 00070  MOVZWL  TOKENLEN, R6                           ; 4385
                                50        04    A7   D0 00073  MOVL    TOKENPTR, R0                            ;
                20        20    60              56   2C 00077  MOVC5   R6, (R0), #32, #32, OLDUSERNAME         ;
                                      08        A8        0007C                                               ;
                          04    A8              56   D0 0007E  MOVL    R6, OLDUSERLEN                          ; 4386
                                68              01   90 00082  MOVB    #1, RENAME_PH2                          ; 4387
                                5C        08    AC   E8 00085  5$:     BLBS    PERMANENT_OK, 12$               ; 4391
                                51      0118    C9   9A 00089  MOVZBL  DEFUSER, R1                            ; 4394
                                50        04    A7   D0 0008E  MOVL    TOKENPTR, R0                            ;
                67        20    0119    C9        51   2D 00092  CMPC5   R1, DEFUSER+1, #32, TOKENLEN, (R0)   ;
                                                60        00099                                               ;
                                                1A   12 0009A  BNEQ    8$                                      ;
                                03              6C   91 0009C  CMPB    (AP), #3                        ; 4396
                                                05   1F 0009F  BLSSU   6$                                      ;
                                      0C        AC   D5 000A1  TSTL    12(AP)                                  ;
                                                08   12 000A4  BNEQ    7$                                      ;
                           00000000G            8F   DD 000A6  6$:     PUSHL   #UAF$_REMDEF                    ; 4398
                                                33   11 000AC  BRB     11$                                     ;
                           00000000G            8F   DD 000AE  7$:     PUSHL   #UAF$_RENDEF                    ; 4400
```

```
                                                 2B 11 000B4             BRB     11$                                                     ;
                                          51     C9 9A 000B6 8$:         MOVZBL  SYSUSER, R1                                             ; 4402
                                          50 04  A7 D0 000BB             MOVL    TOKENPTR, R0                                            ;
                67          20    0151  C9     51 2D 000BF             CMPC5   R1, SYSUSER+1, #32, TOKENLEN, (R0)                      ;
                                                 60 000C6                                                                                ;
                                                 1C 12 000C7             BNEQ    12$                                                     ;
                                          03     6C 91 000C9             CMPB    (AP), #3                                                ; 4405
                                                 05 1F 000CC             BLSSU   9$                                                      ;
                                       0C        AC D5 000CE             TSTL    12(AP)                                                  ;
                                                 08 12 000D1             BNEQ    10$                                                     ;
                                    00000000G    8F DD 000D3 9$:         PUSHL   #UAF$_REMSYS                                            ; 4407
                                                 06 11 000D9             BRB     11$                                                     ;
                                    00000000G    8F DD 000DB 10$:        PUSHL   #UAF$_RENSYS                                            ; 4409
                                          6A     01 FB 000E1 11$:        CALLS   #1, LIB$SIGNAL                                          ;
                                                 04 000E4                RET                                                             ;
                                       04        AC DD 000E5 12$:        PUSHL   LOCK_RECORD                                             ; 4413
                                       04        A7 DD 000E8             PUSHL   TOKENPTR                                                ;
                                    7E           67 3C 000EB             MOVZWL  TOKENLEN, -(SP)                                         ;
                          00000000V 00           03 FB 000EE             CALLS   #3, LOCATE_USER                                         ;
                                    04           50 E9 000F5             BLBC    R0, 13$                                                 ;
                                    50           01 D0 000F8             MOVL    #1, R0                                                  ; 4414
                                                 04 000FB                RET                                                             ;
                                    50     18    A7 D0 000FC 13$:        MOVL    RMSERR, R0                                              ; 4416
                          000182B2  8F           50 D1 00100             CMPL    R0, #98994                                              ;
                                                 13 12 00107             BNEQ    14$                                                     ;
                                       04        A7 DD 00109             PUSHL   TOKENPTR                                                ; 4418
                                    7E           67 3C 0010C             MOVZWL  TOKENLEN, -(SP)                                         ;
                                                 02 DD 0010F             PUSHL   #2                                                      ;
                                    00000000G    8F DD 00111             PUSHL   #UAF$_BADUSR                                            ;
                                          6A     04 FB 00117             CALLS   #4, LIB$SIGNAL                                          ;
                                                 0D 11 0011A             BRB     15$                                                     ;
                                                 50 DD 0011C 14$:        PUSHL   R0                                                      ; 4420
                                                 7E D4 0011E             CLRL    -(SP)                                                   ;
                                    00000000G    8F DD 00120             PUSHL   #UAF$_GETERR                                            ;
                                          6A     03 FB 00126             CALLS   #3, LIB$SIGNAL                                          ;
                                                 50 D4 00129 15$:        CLRL    R0                                                      ; 4422
                                                 04 0012B                RET                                                             ;

; Routine Size:  300 bytes,    Routine Base:  $CODE$ + 25D3
```

```
;  4366          4423  1 %sbttl 'locate_user - lookup user record in UAF'
;  4367          4424  1 routine locate_user (size, buffer, lock_record) =
;  4368          4425  2 begin
;  4369          4426  2
;  4370          4427  2 !++
;  4371          4428  2 !
;  4372          4429  2 ! FUNCTIONAL DESCRIPTION:
;  4373          4430  2 !
;  4374                2 !       Routine to locate a user record in the UAF file.
;  4375          4432  2 !
;  4376          4433  2 ! INPUTS:
;  4377          4434  2 !
;  4378          4435  2 !       SIZE - size of the username string
;  4379          4436  2 !       BUFFER - address of the username string
;  4380          4437  2 !       LOCK_RECORD - specifies that the GET shall lock the record
;  4381          4438  2 !
;  4382          4439  2 ! IMPLICIT INPUTS:
;  4383          4440  2 !
;  4384          4441  2 !       UAFRAB - RMS data structure for SYSUAF.DAT
;  4385          4442  2 !
;  4386          4443  2 ! OUTPUTS:
;  4387          4444  2 !
;  4388          4445  2 !       none
;  4389          4446  2 !
;  4390          4447  2 ! IMPLICIT OUTPUTS:
;  4391          4448  2 !
;  4392          4449  2 !       If record is found, RECBUF contains the located record.
;  4393          4450  2 !
;  4394          4451  2 ! ROUTINE VALUE:
;  4395          4452  2 !
;  4396          4453  2 !       true -> record found
;  4397          4454  2 !       false -> record not found
;  4398          4455  2 !
;  4399          4456  2 ! SIDE EFFECTS:
;  4400          4457  2 !
;  4401          4458  2 !       none
;  4402          4459  2 !--
;  4403          4460  2
;  4404          4461  2 local
;  4405          4462  2         found;                                  ! record found indicator
;  4406          4463  2
;  4407          4464  2 found = true;                                   ! assume record was found
;  4408          4465  2
;  4409          4466  2 ch$copy (.size, .buffer, %char (' '),
;  4410          4467  2          uaf$s_username, recbuf[uaf$t_username]);
;  4411          4468  2
;  4412          4469  2 if .lock_record
;  4413          4470  2 then uafrab[rab$l_rop] = rab$m_rlk
;  4414          4471  2 else uafrab[rab$l_rop] = rab$m_rrl or rab$m_nlk;
;  4415          4472  2
;  4416          4473  3 if not (rmserr = get_uaf_record ())
;  4417          4474  2 then found = false;
;  4418          4475  2
;  4419          4476  2 return .found;                                  ! return indicator
;  4420          4477  1 end;
```

```
                                                00FC 00000 LOCATE_USER:
                                                           .WORD   Save R2,R3,R4,R5,R6,R7                  ; 4424
                                57 00000000'    00  9E 00002           MOVAB   UAFRAB+4, R7
                                56              01  D0 00009           MOVL    #1, FOUND                   ; 4464
                20          20      08  BC          04  AC  2C 0000C   MOVC5   SIZE, @BUFFER, #32, #32, RECBUF+4 ; 4467
                                                FA18    C7      00013
                                09              0C  AC  E9 00016       BLBC    LOCK_RECORD 1$              ; 4469
                                67 00080000     8F  D0 0001A           MOVL    #524288, UAFRAB+4           ; 4470
                                                07  11 00021           BRB     2$
                                67 00100008     8F  D0 00023 1$:       MOVL    #1048584, UAFRAB+4          ; 4471
                    00000000V   00              00  FB 00026 2$:       CALLS   #0, GET_UAF_RECORD          ; 4473
                        FA0C    C7              50  D0 00031           MOVL    R0, RMSERR
                                02              50  E8 00036           BLBS    R0, 3$
                                                56  D4 00039           CLRL    FOUND                       ; 4474
                                50              56  D0 0003B 3$:       MOVL    FOUND, R0                   ; 4476
                                                    04 0003E           RET                                 ; 4477

; Routine Size:  63 bytes,    Routine Base:  $CODE$ + 26FF
```

```
;  4422          4478  1 %sbttl 'get_uaf_record - routine to deal with record locking'
;  4423          4479  1 routine get_uaf_record =
;  4424          4480  2 begin
;  4425          4481  2
;  4426          4482  2 !++
;  4427          4483  2 !
;  4428          4484  2 ! FUNCTIONAL DESCRIPTION:
;  4429          4485  2 !
;  4430          4486  2 !       A common routine to GET records from SYSUAF.DAT, deal with retries
;  4431          4487  2 !       when the record is locked.
;  4432          4488  2 !
;  4433          4489  2 ! INPUTS:
;  4434          4490  2 !
;  4435          4491  2 !       none
;  4436          4492  2 !
;  4437          4493  2 ! IMPLICIT INPUTS:
;  4438          4494  2 !
;  4439          4495  2 !       UAFRAB - RMS data structure for SYSUAF.DAT
;  4440          4496  2 !
;  4441          4497  2 ! OUTPUTS:
;  4442          4498  2 !
;  4443          4499  2 !       none
;  4444          4500  2 !
;  4445          4501  2 ! IMPLICIT OUTPUTS:
;  4446          4502  2 !
;  4447          4503  2 !       RECBUF - The user's record
;  4448          4504  2 !
;  4449          4505  2 ! ROUTINE VALUE:
;  4450          4506  2 !
;  4451          4507  2 !       RMS status code
;  4452          4508  2 !
;  4453          4509  2 ! SIDE EFFECTS:
;  4454          4510  2 !
;  4455          4511  2 !       none
;  4456          4512  2 !--
;  4457          4513  2
;  4458          4514  2 local
;  4459          4515  2         counter,                              ! number of retries remaining
;  4460          4516  2         success;
;  4461          4517  2
;  4462          4518  2 !
;  4463          4519  2 ! If anybody's record is locked it shouldn't remain that way for long.
;  4464          4520  2 ! Also, ignore the special system password record.
;  4465          4521  2 !
;  4466          4522  2
;  4467          4523  2 counter = retry_rlk;
;  4468          4524  4 while ( ((success = $get (rab = uafrab)) eql rms$_rlk)
;  4469          4525  3         or ch$eql(UAF$S_USERNAME, RECBUF[UAF$T_USERNAME],
;  4470          4526  3                   17, uplit(%ascii'<System-Password>'), %c' ') )
;  4471          4527  3       and ( (counter = .counter - 1) geq 0)
;  4472          4528  2 do
;  4473          4529  2     if $schdwk (daytim = wakedelta) then $hiber;
;  4474          4530  2
;  4475          4531  2 .success
;  4476          4532  1 end;
```

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

72 6F 77 73 73 61 50 2B 6D 65 74 73 79 53 3C  00890 P.AFJ:    .ASCII  \<System+Password>\<0><0><0>                    ;
                                              00 00 00 3E 64  0089F

                                                                .PSECT  $CODE$,NOWRT,2

                                                        003C 00000 GET_UAF_RECORD:
                                                                .WORD   Save R2,R3,R4,R5                        ; 4479
                                                 54             MOVL    #8, COUNTER                            ; 4523
                                                    00000000'   PUSHAB  UAFRAB                                 ; 4524
                                    00000000G    00             CALLS   #1, SYS$GET
                                                 55             MOVL    R0, SUCCESS
                                    000182AA     8F             CMPL    SUCCESS, #98986
                                                                BEQL    2$
                 11                20 00000000'  00             CMPC5   #32, RECBUF+4, #32, #17, P.AFJ          ; 4525
                                                    00000000'   BNEQ    3$
                                                                DECL    COUNTER                                ; 4527
                                                                BLSS    3$
                                                                CLRL    -(SP)                                  ; 4529
                                                    00000000'   PUSHAB  WAKEDELTA
                                                                CLRQ    -(SP)
                                    00000000G    00             CALLS   #4, SYS$SCHDWK
                                                 BF             BLBC    R0, 1$
                                    00000000G    00             CALLS   #0, SYS$HIBER
                                                                BRB     1$
                                                 50             MOVL    SUCCESS, R0                            ; 4532
                                                                RET                                            ;

; Routine Size:  83 bytes,    Routine Base:  $CODE$ + 273E
```

```
;  4478          4533  1 %sbttl 'get_cmd_line - input user command line'
;  4479          4534  1 routine get_cmd_line =
;  4480          4535  2 begin
;  4481          4536  2
;  4482          4537  2 !++
;  4483          4538  2 !
;  4484          4539  2 ! FUNCTIONAL DESCRIPTION:
;  4485          4540  2 !
;  4486          4541  2 !       This routine reads in the command line from the user,
;  4487          4542  2 !       reading additional lines if continuation is specified by
;  4488          4543  2 !       a '-' as the last character on the input line.
;  4489          4544  2 !       A zero byte is inserted following the last input character read.
;  4490          4545  2 !
;  4491          4546  2 ! INPUTS:
;  4492          4547  2 !
;  4493          4548  2 !       none
;  4494          4549  2 !
;  4495          4550  2 ! IMPLICIT INPUTS:
;  4496          4551  2 !
;  4497          4552  2 !       CMDBUF - buffer to receive the user's command line
;  4498          4553  2 !       CMDBUFLEN - literal length of CMDBUF
;  4499          4554  2 !
;  4500          4555  2 ! OUTPUTS:
;  4501          4556  2 !
;  4502          4557  2 !       none
;  4503          4558  2 !
;  4504          4559  2 ! IMPLICIT OUTPUTS:
;  4505          4560  2 !
;  4506          4561  2 !       CMDBUF is filled with command line
;  4507          4562  2 !
;  4508          4563  2 ! ROUTINE VALUE:
;  4509          4564  2 !
;  4510          4565  2 !       none
;  4511          4566  2 !
;  4512          4567  2 ! SIDE EFFECTS:
;  4513          4568  2 !
;  4514          4569  2 !       none
;  4515          4570  2 !--
;  4516          4571  2
;001 **NEW**     4572  2     local
;002 **NEW**     4573  2         STATUS;
;003 **NEW**     4574  2
;004 **NEW**     4575  2     do
;005 **NEW**     4576  2         STATUS = LIB$GET_INPUT(CMDLINDSC, %ascid'UAF> ')
;006 **NEW**     4577  2     until CMDLINDSC[DSC$W_LENGTH] neq 0 or .STATUS eql RMS$_EOF;
;007 **NEW**     4578  2
;008 **NEW**     4579  2     if .STATUS eql RMS$_EOF then
;009 **NEW**     4580  2         exit_uaf();
;010 **NEW**     4581  2
;011 **NEW**     4582  2     return .STATUS;
;  4569-52       4583  2
;  4570          4584  1 end;


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

```
                    00  00  00  20  3E  46  41  55  008A4 P.AFL:  .ASCII  \UAF> \<0><0><0>
                                    010E0005  008AC P.AFK:  .LONG   17694725
                                    00000000'  008B0         .ADDRESS P.AFL


                                                             .PSECT  $CODE$,NOWRT,2

                                                 000C 00000 GET_CMD_LINE:
                                                             .WORD   Save R2,R3                  ; 4534
                                  53 00000000'   00   9E 00002          MOVAB   CMDLINDSC, R3
                                     00000000'   00   9F 00009 1$:      PUSHAB  P.AFK               ; 4576
                                                 53   DD 0000F          PUSHL   R3
                        00000000G   00           02   FB 00011          CALLS   #2, LIB$GET_INPUT
                                    52           50   D0 00018          MOVL    R0, STATUS
                                    50           63   9E 0001B          MOVAB   CMDLINDSC, R0       ; 4577
                                                 09   12 0001E          BNEQ    2$
                        0001827A    8F           52   D1 00020          CMPL    STATUS, #98938
                                                 E0   12 00027          BNEQ    1$
                        0001827A    8F           52   D1 00029 2$:      CMPL    STATUS, #98938      ; 4579
                                                 07   12 00030          BNEQ    3$
                        00000000V   00           00   FB 00032          CALLS   #0, EXIT_UAF        ; 4580
                                    50           52   D0 00039 3$:      MOVL    STATUS, R0          ; 4582
                                                 04   0003C            RET                         ; 4584

; Routine Size.  61 bytes,    Routine Base:  $CODE$ + 2791
```

```
;  4572     4585  1 %sbttl 'ask - prompt terminal for input'
;  4573     4586  1 global routine ask (string, buffer, len) : novalue =
;  4574     4587  2 begin
;  4575     4588  2
;  4576     4589  2 !++
;  4577     4590  2 !
;  4578     4591  2 ! FUNCTIONAL DESCTIPTION:
;  4579     4592  2 !
;  4580     4593  2 !       Routine to prompt user for input.  The input string is read
;  4581     4594  2 !       into the user specified buffer and the size read
;  4582     4595  2 !       is placed in the global INSIZE.
;  4583     4596  2 !
;  4584     4597  2 ! INPUTS:
;  4585     4598  2 !
;  4586     4599  2 !       STRING - the address of a counted ascii prompt string
;  4587     4600  2 !       BUFFER - address of the input buffer
;  4588     4601  2 !       LEN - length of the input buffer
;  4589     4602  2 !
;  4590     4603  2 ! IMPLICIT INPUTS:
;  4591     4604  2 !
;  4592     4605  2 !       none
;  4593     4606  2 !
;  4594     4607  2 ! OUTPUTS:
;  4595     4608  2 !
;  4596     4609  2 !       none
;  4597     4610  2 !
;  4598     4611  2 ! IMPLICIT OUTPUTS:
;  4599     4612  2 !
;  4600     4613  2 !       INSIZE - size of the input string
;  4601     4614  2 !
;  4602     4615  2 ! ROUTINE VALUE:
;  4603     4616  2 !
;  4604     4617  2 !       none
;  4605     4618  2 !
;  4606     4619  2 ! SIDE EFFECTS:
;  4607     4620  2 !
;  4608     4621  2 !       none
;  4609     4622  2 !--
;  4610     4623  2
;  4611     4624  2 map
;  4612     4625  2         buffer  : ref vector[,byte];
;  4613     4626  2
;  4614     4627  2 inrab[rab$l_pbf] = .string + 1;                    ! prompt string address
;  4615     4628  2 inrab[rab$b_psz] = . (.string)<0,8>;               ! prompt size
;  4616     4629  2 inrab[rab$l_ubf] = .buffer;                        ! buffer address
;  4617     4630  2 inrab[rab$w_usz] = .len;                           ! buffer size
;  4618     4631  2
;  4619     4632  2 !
;  4620     4633  2 ! If end of file encountered on get (either ^Z from terminal or
;  4621     4634  2 ! end of file on indirect command file) then take exit path.
;  4622     4635  2 !
;  4623     4636  2
;  4624     4637  2 if $get (rab=inrab) eql rms$_eof
;  4625     4638  2 then exit_uaf ();
;  4626     4639  2
;  4627     4640  2 if (insize = .inrab[rab$w_rsz]) neq 0                ! get input size
;  4628     4641  2 then
```

```
;                                ask - prompt terminal for input

; 4629          4642  3     begin
; 4630          4643  3     incru i to .insize - 1                          ! Upcasing is done here because the CVT
; 4631          4644  3     do                                              ! option does not work under batch.
; 4632          4645  4         begin
; 4633          4646  4         if .buffer[.i] gequ 'a' and .buffer[.i] lequ 'z'
; 4634          4647  4         then buffer[.i] = .buffer[.i] and not %o'040';
; 4635          4648  3         end;
; 4636          4649  2     end;
; 4637          4650  1 end;


                                                0004 00000          .ENTRY  ASK, Save R2                                    ; 4586
                                52 00000000'    00   9E 00002          MOVAB   INSIZE, R2                                      ;
                    0090    C2          04      AC   01   C1 00009          ADDL3   #1, STRING, INRAB+48                            ; 4627
                            0094    C2          04   BC   90 00010          MOVB    @STRING, INRAB+52                               ; 4628
                            0084    C2          08   AC   D0 00016          MOVL    BUFFER, INRAB+36                                ; 4629
                            0080    C2          0C   AC   B0 0001C          MOVW    LEN, INRAB+32                                   ; 4630
                                                60   A2   9F 00022          PUSHAB  INRAB                                           ; 4637
                        00000000G   00               01   FB 00025          CALLS   #1, SYS$GET                                     ;
                        0001827A    8F               50   D1 0002C          CMPL    R0, #98938                                      ;
                                                     07   12 00033          BNEQ    1$                                              ;
                        00000000V   00               00   FB 00035          CALLS   #0, EXIT_UAF                                    ; 4638
                                    62      0082     C2   3C 0003C 1$:      MOVZWL  INRAB+34, INSIZE                                ; 4640
                                                     24   13 00041          BEQL    5$                                              ;
                            51              62           01   C3 00043          SUBL3   #1, INSIZE, R1                                  ; 4643
                                                         50   D4 00047          CLRL    I                                               ;
                                                         17   11 00049          BRB     4$                                              ;
                        61      8F          08 BC40      91   0004B 2$:      CMPB    @BUFFER[I], #97                                 ; 4646
                                                         0D   1F 00051          BLSSU   3$                                              ;
                        7A      8F          08 BC40      91   00053          CMPB    @BUFFER[I], #122                                ;
                                                         05   1A 00059          BGTRU   3$                                              ;
                    08 BC40                              20   8A 0005B          BICB2   #32, @BUFFER[I]                                 ; 4647
                                                         50   D6 00060 3$:      INCL    I                                               ; 4643
                            51                           50   D1 00062 4$:      CMPL    I, R1                                           ;
                                                         E4   1B 00065          BLEQU   2$                                              ;
                                                              04 00067 5$:      RET                                                     ; 4650

; Routine Size:  104 bytes,    Routine Base:  $CODE$ + 27CE
```

```
;  4639          4651  1 %sbttl 'fmt_sys_msg - output system message file message'
;  4640          4652  1 global routine fmt_sys_msg (faostr, msgid, p1) : novalue =
;  4641          4653  2 begin
;  4642          4654  2
;  4643          4655  2 !++
;  4644          4656  2 !
;  4645          4657  2 ! FUNCTIONAL DESCRIPTION:
;  4646          4658  2 !
;  4647          4659  2 !       This routine outputs an error message followed by
;  4648          4660  2 !       the text found in the system message file for the
;  4649          4661  2 !       error condition.  If the message is not found, the message
;  4650          4662  2 !       ID itself is printed.
;  4651          4663  2 !
;  4652          4664  2 ! INPUTS:
;  4653          4665  2 !
;  4654          4666  2 !       FAOSTR - address of counted ascii message to be printed
;  4655          4667  2 !       MSGID  - error number
;  4656          4668  2 !       P1     - the first of possibly several parameters to FAO
;  4657          4669  2 !
;  4658          4670  2 ! IMPLICIT INPUTS:
;  4659          4671  2 !
;  4660          4672  2 !       None
;  4661          4673  2 !
;  4662          4674  2 ! OUTPUTS:
;  4663          4675  2 !
;  4664          4676  2 !       None
;  4665          4677  2 !
;  4666          4678  2 ! IMPLICIT OUTPUTS.
;  4667          4679  2 !
;  4668          4680  2 !       None
;  4669          4681  2 !
;  4670          4682  2 ! ROUTINE VALUE:
;  4671          4683  2 !
;  4672          4684  2 !       None
;  4673          4685  2 !
;  4674          4686  2 ! SIDE EFFECTS:
;  4675          4687  2 !
;  4676          4688  2 !       None
;  4677          4689  2 !--
;  4678          4690  2
;  4679          4691  2 local
;  4680          4692  2         buffer  : vector [200, byte],           ! buffer to receive message
;  4681          4693  2         bufdsc  : vector [2, long],             ! string descriptor
;  4682          4694  2         code;                                   ! save return code
;  4683          4695  2
;  4684          4696  2 bufdsc[0] = 200;                                ! construct string descriptor
;  4685          4697  2 bufdsc[1] = buffer;
;  4686          4698  2
;  4687          4699  2 code = $getmsg (msgid = .msgid, msglen = bufdsc[0], bufadr = bufdsc[0]);
;  4688          4700  2
;  4689          4701  2 !
;  4690          4702  2 ! Output internal message.  Then output system error or error number.
;  4691          4703  2 !
;  4692          4704  2 faodsc[dsc$w_length] = . (.faostr)<0,8>;         ! input string descriptor
;  4693          4705  2 faodsc[dsc$a_pointer] = .faostr+1;
;  4694        P 4706  2 $faol (ctrstr = faodsc,
;  4695        P 4707  2         outlen = outrab[rab$w_rsz],
```

```
;  4696      P 4708  2          outbuf = disdsc,
;  4697        4709  2          prmlst = p1);
;  4698        4710  2
;  4699        4711  2 $put (rab = outrab);
;  4700        4712  2
;  4701        4713  2 if .code eql ss$_msgnotfnd
;  4702        4714  2 then LIB$SIGNAL(UAF$_SYSMSG2, 1, .msgid)
;  4703        4715  2 else LIB$SIGNAL(UAF$_SYSMSG1, 1, bufdsc[0]);
;  4704        4716  1 end;


                                                                .EXTRN  SYS$GETMSG, SYS$FAOL

                                                    000C 00000  .ENTRY  FMT_SYS_MSG, Save R2,R3                 ; 4652
                                          53 00000000'  00  9E  MOVAB  FAODSC, R3                               ;
                                             5E     FF34  CE  9E  MOVAB  -204(SP), SP                          ;
                                             7E       C8  8F  9A  MOVZBL #200, BUFDSC                          ; 4696
                                       04    AE       08  AE  9E  MOVAB  BUFFER, BUFDSC+4                      ; 4697
                                             7E           0F  7D  MOVQ   #15, -(SP)                            ; 4699
                                                      08  AE  9F  PUSHAB BUFDSC                               ;
                                                      0C  AE  9F  PUSHAB BUFDSC                               ;
                                                      08  AC  DD  PUSHL  MSGID                                 ;
                                          00000000G   00      05  FB  CALLS  #5, SYS$GETMSG                     ;
                                                      52      50  D0  MOVL   R0, CODE                           ;
                                                      63      04  BC  MOVZBW @FAOSTR, FAODSC                   ; 4704
                                    04    A3      04    AC    01  C1  ADDL3  #1, FAOSTR, FAODSC+4              ; 4705
                                                      0C  AC  9F  PUSHAB P1                                   ; 4709
                                                      F8  A3  9F  PUSHAB DISDSC                               ;
                                                    072E  C3  9F  PUSHAB OUTRAB+34                            ;
                                                      53  DD  PUSHL  R3                                       ;
                                          00000000G   00      04  FB  CALLS  #4, SYS$FAOL                      ;
                                                    070C  C3  9F  PUSHAB OUTRAB                               ; 4711
                                          00000000G   00      01  FB  CALLS  #1, SYS$PUT                        ;
                                          00000621    8F      52  D1  CMPL   CODE, #1569                       ; 4713
                                                      0D  12  BNEQ   1$                                       ;
                                                   08     AC  DD  PUSHL  MSGID                               ; 4714
                                                      01  DD  PUSHL  #1                                       ;
                                                00000000G 8F  DD  PUSHL  #UAF$_SYSMSG2                        ;
                                                      0A  11  BRB    2$                                       ;
                                                      5E  DD  PUSHL  SP                 1$:                   ; 4715
                                                      01  DD  PUSHL  #1                                       ;
                                                00000000G 8F  DD  PUSHL  #UAF$_SYSMSG1                        ;
                                          00000000G   00      03  FB  CALLS  #3, LIB$SIGNAL          2$:       ;
                                                          04  0007C RET                                        ; 4716

; Routine Size:  125 bytes,    Routine Base:  $CODE$ + 2836
```

```
;  4706     4717  1 %sbttl 'faoout - output formatted message'
;  4707     4718  1 global routine faoout (string, p1) =
;  4708     4719  2 begin
;  4709     4720  2
;  4710     4721  2 !++
;  4711     4722  2 !
;  4712     4723  2 ! FUNCTIONAL DESCRIPTION:
;  4713     4724  2 !
;  4714     4725  2 !       Routine to output a formatted string.
;  4715     4726  2 !
;  4716     4727  2 ! INPUTS:
;  4717     4728  2 !
;  4718     4729  2 !       STRING - address of a counted ASCII FAO control string.
;  4719     4730  2 !       P1 - the first of possibly several parameters to FAO
;  4720     4731  2 !
;  4721     4732  2 ! IMPLICIT INPUTS:
;  4722     4733  2 !
;  4723     4734  2 !       none
;  4724     4735  2 !
;  4725     4736  2 ! OUTPUTS:
;  4726     4737  2 !
;  4727     4738  2 !       none
;  4728     4739  2 !
;  4729     4740  2 ! IMPLICIT OUTPUTS:
;  4730     4741  2 !
;  4731     4742  2 !       none
;  4732     4743  2 !
;  4733     4744  2 ! ROUTINE VALUE:
;  4734     4745  2 !
;  4735     4746  2 !       FAOOUT always returns FALSE, as it is often used on the
;  4736     4747  2 !       return from an error condition.
;  4737     4748  2 !
;  4738     4749  2 ! SIDE EFFECTS:
;  4739     4750  2 !
;  4740     4751  2 !       none
;  4741     4752  2 !--
;  4742     4753  2
;  4743     4754  2
;  4744     4755  2 faodsc[dsc$w_length] = . (.string)<0,8>;         ! input string descriptor
;  4745     4756  2 faodsc[dsc$a_pointer] = .string+1;
;  4746   P 4757  2 $faol (ctrstr = faodsc,
;  4747   P 4758  2         outlen = outrab[rab$w_rsz],
;  4748   P 4759  2         outbuf = disdsc,
;  4749     4760  2         prmlst = p1);
;  4750     4761  2
;  4751     4762  2 $put (rab = outrab);
;  4752     4763  2
;  4753     4764  2 false
;  4754     4765  1 end;


                                  0004 00000          .ENTRY  FAOOUT, Save R2                 ; 4718
                52 00000000'   00   9E 00002          MOVAB   FAODSC, R2                      ;
                62         04   BC   9B 00009          MOVZBW  @STRING, FAODSC                 ; 4755
```

```
                04   A2        04   AC                01  C1 0000D        ADDL3   #1, STRING, FAODSC+4                ; 4756
                                                    08  AC  9F 00013        PUSHAB  P1                                  ; 4760
                                                    F8  A2  9F 00016        PUSHAB  DISDSC                              ;
                                                 072E   C2  9F 00019        PUSHAB  OUTRAB+34                           ;
                                                        52  DD 0001D        PUSHL   R2                                  ;
                    00000000G   00                      04  FB 0001F        CALLS   #4, SYS$FAOL                        ;
                                                 070C   C2  9F 00026        PUSHAB  OUTRAB                              ; 4762
                    00000000G   00                      01  FB 0002A        CALLS   #1, SYS$PUT                         ;
                                                        50  D4 00031        CLRL    R0                                  ; 4765
                                                            04 00033        RET                                         ;

; Routine Size:  52 bytes,    Routine Base:  $CODE$ + 28B3
```

```
;  4756          4766  1 %sbttl 'help_uaf - help routine'
;  4757          4767  1 global routine help_uaf : novalue =
;  4758          4768  2 begin
;  4759          4769  2
;  4760          4770  2 !++
;  4761          4771  2 !
;  4762          4772  2 ! FUNCTIONAL DESCRIPTION:
;  4763          4773  2 !
;  4764          4774  2 !       Print out the help message or messages.
;  4765          4775  2 !
;  4766          4776  2 ! INPUTS:
;  4767          4777  2 !
;  4768          4778  2 !       none
;  4769          4779  2 !
;  4770          4780  2 ! OUTPUTS:
;  4771          4781  2 !
;  4772          4782  2 !       none
;  4773          4783  2 !
;  4774          4784  2 ! ROUTINE VALUE:
;  4775          4785  2 !
;  4776          4786  2 !       none
;  4777          4787  2 !
;  4778          4788  2 ! SIDE EFFECTS:
;  4779          4789  2 !
;  4780          4790  2 !       none
;  4781          4791  2 !--
;  4782          4792  2
;  4783          4793  2 map
;  4784          4794  2         cmdlindsc: vector;
;  4785          4795  2
;  4786          4796  2 local
;  4787          4797  2         line_dsc        : vector [2];
;  4788          4798  2
;  4789          4799  2 line_dsc[0] = .cmdlindsc[0];
;  4790          4800  2 line_dsc[1] = .cmdlindsc[1];
;  4791          4801  2
;  4792          4802  2 !
;  4793          4803  2 ! The first thing to do is to remove 'help' from the command line
;  4794          4804  2 ! and give the help routine the remainder of the command line.
;  4795          4805  2 ! Find the beginning of the word 'help'.
;  4796          4806  2 !
;  4797          4807  2 if .(.line_dsc[1])<0,8> eql ' '
;  4798          4808  2 then
;  4799          4809  3     begin
;  4800          4810  3     line_dsc[1] = ch$find_not_ch (.line_dsc[0], .line_dsc[1], ' ');
;  4801          4811  3     line_dsc[0] = .line_dsc[0] - (.line_dsc[1] - .cmdlindsc[1]);
;  4802          4812  2     end;
;  4803          4813  2
;  4804          4814  2 !
;  4805          4815  2 ! Now skip past the 'help' to the first blank (if there is one)
;  4806          4816  2 !
;  4807          4817  2 if ch$fail (line_dsc[1] = ch$find_ch (.line_dsc[0], .line_dsc[1], ' '))
;  4808          4818  2 then
;  4809          4819  2     !
;  4810          4820  2     ! No blank, set empty string
;  4811          4821  2     !
;  4812          4822  2     line_dsc[0] = 0
```

```
;  4813          4823  2 else
;  4814          4824  2     !
;  4815          4825  2     ! Found a blank, set pointer to it
;  4816          4826  2     !
;  4817          4827  2     line_dsc[0] = .cmdlindsc[0] - (.line_dsc[1] - .cmdlindsc[1]);
;  4818          4828  2
;  4819          4829  2 !
;  4820          4830  2 ! Start an interactive HELP session
;  4821          4831  2 !
;  4822          4832  2
;  4823          4833  2 if not lbr$output_help (lib$put_output, 0, line_dsc,
;  4824          4834  2                         $descriptor ('uafhelp'), 0,
;  4825          4835  2                         lib$get_input)
;  4826          4836  2 then LIB$SIGNAL(UAF$_HELPERR);
;  4827          4837  2
;  4828          4838  1 end;


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                70  6C  65  68  66  61  75  008B4 P.AFN:  .ASCII  \uafhelp\                        ;
                                                    008BB       .BLKB   1
                                          00000007  008BC P.AFM:  .LONG   7                                       ;
                                          00000000' 008C0        .ADDRESS P.AFN                                   ;


                                                        .PSECT  $CODE$,NOWRT,2

                                                    000C 00000      .ENTRY  HELP_UAF, Save R2,R3                    ; 4767
                                              5E        04  C2 00002      SUBL2   #4, SP
                                              53 00000000'  00  D0 00005      MOVL    CMDLINDSC, R3                 ; 4799
                                                        53  DD 0000C      PUSHL   R3
                                              52 00000000'  00  D0 0000E      MOVL    CMDLINDSC+4, R2               ; 4800
                                        04    AE            52  D0 00015      MOVL    R2, LINE_DSC+4
                                        20          04      BE  91 00019      CMPB    @LINE_DSC+4, #32              ; 4807
                                                            15  12 0001D      BNEQ    2$
                                  04    BE          6E        20  3B 0001F      SKPC    #32, LINE_DSC, @LINE_DSC+4    ; 4810
                                                            02  12 00024      BNEQ    1$
                                                            51  D4 00026      CLRL    R1
                                        04    AE            51  D0 00028 1$:  MOVL    R1, LINE_DSC+4
                                  50          52          04    AE  C3 0002C      SUBL3   LINE_DSC+4, R2, R0     ; 4811
                                              6E            50  C0 00031      ADDL2   R0, LINE_DSC
                                  04    BE    6E            20  3A 00034 2$:  LOCC    #32, LINE_DSC, @LINE_DSC+4    ; 4817
                                                            02  12 00039      BNEQ    3$
                                                            51  D4 0003B      CLRL    R1
                                        04    AE            51  D0 0003D 3$:  MOVL    R1, LINE_DSC+4
                                                            04  12 00041      BNEQ    4$
                                                            6E  D4 00043      CLRL    LINE_DSC                      ; 4822
                                                            09  11 00045      BRB     5$
                                  50          52          04    AE  C3 00047 4$:  SUBL3   LINE_DSC+4, R2, R0     ; 4827
                                  6E          50                    53  C1 0004C      ADDL3   R3, R0, LINE_DSC
                                                  00000000G 00  9F 00050 5$:  PUSHAB  LIB$GET_INPUT                 ; 4833
                                                            7E  D4 00056      CLRL    -(SP)
                                                  00000000' 00  9F 00058      PUSHAB  P.AFM                         ; 4834
                                                        0C  AE  9F 0005E      PUSHAB  LINE_DSC                      ; 4833
```

```
                                 7E  D4 00061          CLRL    -(SP)                                   ;
                    00000000G    00  9F 00063          PUSHAB  LIB$PUT_OUTPUT                          ;
                                 06  FB 00069          CALLS   #6, LBR$OUTPUT_HELP                     ;
                                 50  E8 00070          BLBS    R0, 6$                                  ;
                    00000000G    8F  DD 00073          PUSHL   #UAF$_HELPERR                           ; 4836
                    00000000G    01  FB 00079          CALLS   #1, LIB$SIGNAL                          ;
                                 04 00080 6$:          RET                                             ; 4838

; Routine Size:  129 bytes,    Routine Base:  $CODE$ + 28E7
```

```
;  4830     4839  1 %sbttl 'exit_uaf - normal exit routine'
;  4831     4840  1 global routine exit_uaf : novalue =
;  4832     4841  2 begin
;  4833     4842  2
;  4834     4843  2 !++
;  4835     4844  2 !
;  4836     4845  2 ! FUNCTIONAL DESCRIPTION:
;  4837     4846  2 !
;  4838     4847  2 !       Normal exit routine.
;  4839     4848  2 !
;  4840     4849  2 ! INPUTS:
;  4841     4850  2 !
;  4842     4851  2 !       none
;  4843     4852  2 !
;  4844     4853  2 ! OUTPUTS:
;  4845     4854  2 !
;  4846     4855  2 !       none
;  4847     4856  2 !
;  4848     4857  2 ! ROUTINE VALUE:
;  4849     4858  2 !
;  4850     4859  2 !       none
;  4851     4860  2 !
;  4852     4861  2 ! SIDE EFFECTS:
;  4853     4862  2 !
;  4854     4863  2 !       image exits
;  4855     4864  2 !--
;  4856     4865  2
;  4857     4866  2 $close (fab = uaffab);
;  4858     4867  2
;  4859     4868  2 !
;  4860     4869  2 ! Inform user of file modifications.
;  4861     4870  2 !
;  4862     4871  2
;  4863     4872  2 if .modify_flag
;  4864     4873  2 then
;  4865     4874  2 !
;  4866     4875  2 ! File has been modified
;  4867     4876  2 !
;  4868     4877  2     LIB$SIGNAL(UAF$_DONEMSG)            ! tell user all is done
;  4869     4878  2 else
;  4870     4879  2 !
;  4871     4880  2 ! Here, no modifications were made to file.
;  4872     4881  2 !
;  4873     4882  2     LIB$SIGNAL(UAF$_NOMODS);
;  4874     4883  2
;  4875     4884  2 if .netuaf_exists
;  4876     4885  2 then
;  4877     4886  3     begin
;  4878     4887  3     if not .netuaf_modified
;  4879     4888  3     then
;  4880     4889  3         LIB$SIGNAL(UAF$_NAFNOMODS)
;  4881     4890  3     else
;  4882     4891  3         LIB$SIGNAL(UAF$_NAFDONEMSG);
;  4883     4892  2     end;
;  4884     4893  2
;  4885     4894  2 if .rdb_exists
;  4886     4895  2 then
```

```
;  4887        4896  3     begin
;  4888        4897  3     if .rightslist_modified
;  4889        4898  3     then
;  4890        4899  3     !
;  4891        4900  3     ! File has been modified
;  4892        4901  3     !
;  4893        4902  3         LIB$SIGNAL(UAF$_RDBDONEMSG)                ! tell user all is done
;  4894        4903  3     else
;  4895        4904  3     !
;  4896        4905  3     ! Here, no modifications were made to file.
;  4897        4906  3     !
;  4898        4907  3         LIB$SIGNAL(UAF$_RDBNOMODS);
;  4899        4908  2     end ;
;  4900        4909  2
;  4901        4910  2 $exit (code = true);
;  4902        4911  1 end;


                                                        .EXTRN  SYS$EXIT

                                        001C 00000      .ENTRY  EXIT_UAF, Save R2,R3,R4                 ; 4840
                    54 00000000'    00   9E 00002      MOVAB   UAFFAB, R4                              ;
                    53 00000000'    00   9E 00009      MOVAB   NETUAF_EXISTS, R3                       ;
                    52 00000000G    00   9E 00010      MOVAB   LIB$SIGNAL, R2                          ;
                                    54   DD 00017      PUSHL   R4                                      ; 4866
                00000000G   00      01   FB 00019      CALLS   #1, SYS$CLOSE                           ;
                            08  F154  C4   E9 00020      BLBC    MODIFY_FLAG, 1$                         ; 4872
                        00000000G   8F   DD 00025      PUSHL   #UAF$_DONEMSG                           ; 4877
                                    06   11 0002B      BRB     2$                                      ;
                        00000000G   8F   DD 0002D 1$:  PUSHL   #UAF$_NOMODS                           ; 4882
                    62              01   FB 00033 2$:  CALLS   #1, LIB$SIGNAL                         ;
                    16              63   E9 00036      BLBC    NETUAF_EXISTS, 5$                       ; 4884
                    08  F158        C4   E8 00039      BLBS    NETUAF_MODIFIED, 3$                     ; 4887
                        00000000G   8F   DD 0003E      PUSHL   #UAF$_NAFNOMODS                         ; 4889
                                    06   11 00044      BRB     4$                                      ;
                        00000000G   8F   DD 00046 3$:  PUSHL   #UAF$_NAFDONEMSG                        ; 4891
                    62              01   FB 0004C 4$:  CALLS   #1, LIB$SIGNAL                         ;
                    15      04      A3   E9 0004F 5$:  BLBC    RDB_EXISTS, 8$                          ; 4894
                    08      0C      A3   E9 00053      BLBC    RIGHTSLIST_MODIFIED, 6$                 ; 4897
                        00000000G   8F   DD 00057      PUSHL   #UAF$_RDBDONEMSG                        ; 4902
                                    06   11 0005D      BRB     7$                                      ;
                        00000000G   8F   DD 0005F 6$:  PUSHL   #UAF$_RDBNOMODS                        ; 4907
                    62              01   FB 00065 7$:  CALLS   #1, LIB$SIGNAL                         ;
                                    01   DD 00068 8$:  PUSHL   #1                                     ; 4910
                00000000G   00      01   FB 0006A      CALLS   #1, SYS$EXIT                            ;
                                    04   00071        RET                                             ; 4911

; Routine Size:  114 bytes,    Routine Base:  $CODE$ + 2968
```

```
;  4904          4912  1 %sbttl 'SIGNAL_SYNTAX - Report missing qualifier'
;  4905          4913  1 global routine SIGNAL_SYNTAX: novalue =
;  4906          4914  2 begin
;  4907          4915  2
;  4908          4916  2     LIB$SIGNAL(UAF$_ZISQUAL);
;  4909          4917  2
;  4910          4918  1 end;


                                                  0000 00000          .ENTRY  SIGNAL_SYNTAX, Save nothing                ; 4913
                                  00000000G   8F   DD 00002          PUSHL   #UAF$_ZISQUAL                              ; 4916
                        00000000G   00            01   FB 00008          CALLS   #1, LIB$SIGNAL                             ;
                                                  04 0000F          RET                                                ; 4918

; Routine Size:  16 bytes,    Routine Base:  $CODE$ + 29DA
```

```
;  4912          4919  1 %sbttl 'acc$exit - exit and cleanup routine'
;  4913          4920  1 routine acc$exit : novalue =
;  4914          4921  2 begin
;  4915          4922  2
;  4916          4923  2 !++
;  4917          4924  2 !
;  4918          4925  2 ! FUNCTIONAL DESCRIPTION:
;  4919          4926  2 !
;  4920          4927  2 !       Exit on error condition.
;  4921          4928  2 !
;  4922          4929  2 ! INPUTS:
;  4923          4930  2 !
;  4924          4931  2 !       none
;  4925          4932  2 !
;  4926          4933  2 ! OUTPUTS:
;  4927          4934  2 !
;  4928          4935  2 !       none
;  4929          4936  2 !--
;  4930          4937  2
;  4931          4938  2 $exit ();
;  4932          4939  1 end;


                                0000 00000 ACC$EXIT:
                                           .WORD   Save nothing                    ; 4920
                                01  DD 00002  PUSHL   #1                          ; 4938
                    00000000G   01  FB 00004  CALLS   #1, SYS$EXIT                 ;
                                    04 0000B  RET                                  ; 4939

; Routine Size:  12 bytes,    Routine Base:  $CODE$ + 29EA
```

```
;  4934     4940  1 %sbttl 'UAF$MOD_SYS_PWD -- modify the system password'
;  4935     4941  1 global routine UAF$MOD_SYS_PWD: novalue =
;  4936     4942  2 begin
;  4937     4943  2
;  4938     4944  2 !++
;  4939     4945  2 !
;  4940     4946  2 ! Functional Description:
;  4941     4947  2 !
;  4942     4948  2 !       Modify the system password record by doing a $PUT on the UAF
;  4943     4949  2 !       with the UIF bit set for the username "<System+Password>"
;  4944     4950  2 !
;  4945     4951  2 !--
;  4946     4952  2
;  4947     4953  2 local
;  4948     4954  2
;  4949     4955  2     RBF,
;  4950     4956  2     PWD: block[DSC$K_D_BLN, byte]
;  4951     4957  2          preset([DSC$B_CLASS] = DSC$K_CLASS_D),
;  4952     4958  2     ROP;
;  4953     4959  2
;  4954     4960  2 !
;  4955     4961  2 !       Save current settings
;  4956     4962  2 !
;  4957     4963  2
;  4958     4964  2 RBF = .UAFRAB[RAB$L_RBF];
;  4959     4965  2 ROP = .UAFRAB[RAB$L_ROP];
;  4960     4966  2
;  4961     4967  2 !
;  4962     4968  2 !       Get new password and encrypt it
;  4963     4969  2 !
;  4964     4970  2
;  4965     4971  2 CLI$GET_VALUE(%ascid'SYSTEM_PASSWORD', PWD);
;  4966     4972  2
;  4967     4973  2 if .PWD[DSC$W_LENGTH] neq 0 then
;  4968     4974  2     LGI$HPWD(REC_ENCRYPT_DSC, PWD, UAF$C_PURDY_V, 0, %ASCID'<System+Password>')
;  4969     4975  2 else
;  4970     4976  2     ch$fill(0, UAF$S_PWD, RECBUF[UAF$Q_PWD]);
;  4971     4977  2
;  4972     4978  2 !
;  4973     4979  2 !       Fill in fields of the UAF record
;  4974     4980  2 !
;  4975     4981  2
;  4976     4982  2 ch$copy(17, UPLIT('<System+Password>'), ' ',
;  4977     4983  2          UAF$S_USERNAME, RECBUF[UAF$T_USERNAME]);
;  4978     4984  2 RECBUF[UAF$W_SALT] = 0;
;  4979     4985  2
;  4980     4986  2 !
;  4981     4987  2 !       Modify the RAB for our needs
;  4982     4988  2 !
;  4983     4989  2
;  4984     4990  2 UAFRAB[RAB$V_UIF] = 1;
;  4985     4991  2 UAFRAB[RAB$W_RSZ] = UAF$C_LENGTH;
;  4986     4992  2 UAFRAB[RAB$L_RBF] = RECBUF;
;  4987     4993  2
;  4988     4994  2 !
;  4989     4995  2 !       Modify the system password
;  4990     4996  2 !
```

```
;  4991          4997  2
;  4992          4998  2 $PUT(RAB=UAFRAB);
;  4993          4999  2
;  4994          5000  2 !
;  4995          5001  2 !       Replace the settings
;  4996          5002  2 !
;  4997          5003  2
;  4998          5004  2 UAFRAB[RAB$L_RBF] = .RBF;
;  4999          5005  2 UAFRAB[RAB$L_ROP] = .ROP;
;  5000          5006  2
;  5001          5007  1 end;


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

44 52 4F 57 53 53 41 50 5F 4D 45 54 53 59 53  008C4 P.AFP:   .ASCII  \SYSTEM_PASSWORD\<0>                          ;
                                                 00  008D3                                                      ;
                                           010E000F  008D4 P.AFO:   .LONG   17694735                            ;
                                          00000000'  008D8          .ADDRESS P.AFP                              ;
72 6F 77 73 73 61 50 2B 6D 65 74 73 79 53 3C  008DC P.AFR:   .ASCII  \<System+Password>\<0><0><0>                ;
                                       00 00 00 3E 64  008EB                                                    ;
                                           010EC011  008F0 P.AFQ:   .LONG   17694737                            ;
                                          00000000'  008F4          .ADDRESS P.AFR                              ;
72 6F 77 73 73 61 50 2B 6D 65 74 73 79 53 3C  008F8 P.AFS:   .ASCII  \<System+Password>\<0><0><0>                ;
                                       00 00 00 3E 64  00907                                                    ;


                                                                .PSECT  $CODE$,NOWRT,2

                                           03FC 00000           .ENTRY  UAF$MOD_SYS_PWD, Save R2,R3,R4,R5,R6,R7,R8,-  ; 4941
                                                                        R9                                          ;
                                    59 00000000'   00 9E 00002          MOVAB   P.AFO, R9                           ;
                                    58 00000000'   00 9E 00009          MOVAB   UAFRAB+40, R8                       ;
                                    5E             04 C2 00010          SUBL2   #4, SP                              ;
                                       02000000    8F DD 00013          PUSHL   #33554432                           ;  4957
                                             04    AE D4 00019          CLRL    PWD+4                               ;
                                    57             68 D0 0001C          MOVL    UAFRAB+40, RBF                      ;  4964
                                    56           DC A8 D0 0001F         MOVL    UAFRAB+4, ROP                       ;  4965
                                           4200  8F BB 00023          PUSHR   #^M<R9,SP>                          ;  4971
                              00000000G    00       02 FB 00027          CALLS   #2, CLI$GET_VALUE                   ;
                                                    6E B5 0002E          TSTW    PWD                                 ;  4973
                                                    16 13 00030          BEQL    1$                                  ;
                                             1C     A9 9F 00032          PUSHAB  P.AFQ                               ;  4974
                                    7E             02 7D 00035          MOVQ    #2, -(SP)                           ;
                                             0C    AE 9F 00038          PUSHAB  PWD                                 ;
                                           F918    C9 9F 0003B          PUSHAB  REC_ENCRYPT_DSC                     ;
                              00000000G    00       05 FB 0003F          CALLS   #5, LGI$HPWD                        ;
                                                    08 11 00046          BRB     2$                                  ;
08          00                  6E                00 2C 00048 1$:      MOVC5   #0, (SP), #0, #8, RECBUF+340        ;  4976
                                       FB44         C8   0004D                                                      ;
20          20        24        A9                11 2C 00050 2$:      MOVC5   #17, P.AFS, #32, #32, RECBUF+4      ;  4983
                                       F9F4         C8   00056                                                      ;
                                       FB56         C8 B4 00059          CLRW    RECBUF+358                          ;  4984
                                 DC    A8                10 88 0005D    BISB2   #16, UAFRAB+4                       ;  4990
                                 FA    A8          0584  8F B0 00061    MOVW    #1412, UAFRAB+34                    ;  4991
```

```
                            68     F9F0   C8  9E 00067        MOVAB   RECBUF, UAFRAB+40                ; 4992
                              D8   A8  9F 0006C              PUSHAB  UAFRAB                           ; 4998
                00000000G   00     01  FB 0006F              CALLS   #1, SYS$PUT                      ;
                            68     57  D0 00076              MOVL    RBF, UAFRAB+40                   ; 5004
                         DC A8     56  D0 00079              MOVL    ROP, UAFRAB+4                    ; 5005
                                       04 0007D              RET                                      ; 5007

; Routine Size:  126 bytes,    Routine Base:  $CODE$ + 29F6
```

```
;  5003          5008  1 %sbttl 'security_audit - perform a security audit'
;  5004          5009  1 routine security_audit (code, old_user) : novalue =
;  5005          5010  2 begin
;  5006          5011  2
;  5007          5012  2 !++
;  5008          5013  2 !
;  5009          5014  2 ! FUNCTIONAL DESCRIPTION:
;  5010          5015  2 !
;  5011          5016  2 !       Perform a security audit, if needed.
;  5012          5017  2 !
;  5013          5018  2 ! INPUTS:
;  5014          5019  2 !
;  5015          5020  2 !       CODE - Security audit record id
;  5016          5021  2 !       OLD_USER - Old username (optional depending on the record id)
;  5017          5022  2 !
;  5018          5023  2 ! IMPLICIT INPUTS:
;  5019          5024  2 !
;  5020          5025  2 !       PCB_STS - This process's PCB status
;  5021          5026  2 !       UAF$GQ_SYSUAFF - SYSUAF fields modified
;  5022          5027  2 !
;  5023          5028  2 ! OUTPUTS:
;  5024          5029  2 !
;  5025          5030  2 !       none
;  5026          5031  2 !
;  5027          5032  2 ! IMPLICIT OUTPUTS:
;  5028          5033  2 !
;  5029          5034  2 !       none
;  5030          5035  2 !
;  5031          5036  2 ! ROUTINE VALUE:
;  5032          5037  2 !
;  5033          5038  2 !       none
;  5034          5039  2 !
;  5035          5040  2 ! SIDE EFFECTS:
;  5036          5041  2 !
;  5037          5042  2 !       none
;  5038          5043  2 !
;  5039          5044  2 !--
;  5040          5045  2
;  5041          5046  2 external routine
;  5042          5047  2         nsa$event_audit;                    ! Kernel mode auditing routine
;  5043          5048  2
;  5044          5049  2 external
;  5045          5050  2         nsa$gr_alarmvec : block [,byte], ! Security audit alarm vector
;  5046          5051  2         nsa$gr_journvec : block [,byte]; ! Security audit journal vector
;  5047          5052  2
;  5048          5053  2 macro
;  5049        M 5054  2         add_quad_packet(type, a_quad) = ! Add a quadword packet to list
;  5050        M 5055  2             begin
;  5051        M 5056  2             (.arglist_ptr)<0,16> = %name('nsa$k_pkttyp_', type);
;  5052        M 5057  2             arglist_ptr = .arglist_ptr + 2;
;  5053        M 5058  2             (.arglist_ptr)<0,16> = nsa$k_arg_mech_quad;
;  5054        M 5059  2             arglist_ptr = .arglist_ptr + 2;
;  5055        M 5060  2             (.arglist_ptr)<0,32> = .(a_quad)<0,32>;
;  5056        M 5061  2             arglist_ptr = .arglist_ptr + 4;
;  5057        M 5062  2             (.arglist_ptr)<0,32> = .(a_quad)<32,32>;
;  5058        M 5063  2             arglist_ptr = .arglist_ptr + 4;
;  5059        M 5064  2             arglist[nsa$b_arg_pktnum] = .arglist[nsa$b_arg_pktnum] + 1;
```

```
;  5060          5065  2              end%,
;  5061        M 5066  2          add_descr_packet(type, len, ptr) = ! Add a descriptor packet to list
;  5062        M 5067  2              begin
;  5063        M 5068  2              (.arglist_ptr)<0,16> = %name('nsa$k_pkttyp_', type);
;  5064        M 5069  2              arglist_ptr = .arglist_ptr + 2;
;  5065        M 5070  2              (.arglist_ptr)<0,16> = nsa$k_arg_mech_descr;
;  5066        M 5071  2              arglist_ptr = .arglist_ptr + 2;
;  5067        M 5072  2              (.arglist_ptr)<0,32> = len;
;  5068        M 5073  2              arglist_ptr = .arglist_ptr + 4;
;  5069        M 5074  2              (.arglist_ptr)<0,32> = ptr;
;  5070        M 5075  2              arglist_ptr = .arglist_ptr + 4;
;  5071        M 5076  2              arglist[nsa$b_arg_pktnum] = .arglist[nsa$b_arg_pktnum] + 1;
;  5072          5077  2              end%;
;  5073          5078  2
;  5074          5079  2  local
;  5075          5080  2          arglist         : block         ! Argument list for NSA$EVENT_AUDIT
;  5076          5081  2              [nsa$k_arghdr_length + ((2 + 2 + 8) * 6), byte],
;  5077          5082  2          arglist_ptr;                    ! Packet fill in pointer
;  5078          5083  2
;  5079          5084  2  ch$fill(0, %allocation(arglist), arglist); ! Clear out the argument list
;  5080          5085  2
;  5081          5086  2  if .nsa$gr_alarmvec[nsa$v_evt_uaf]      ! If alarm auditing,
;  5082          5087  2  then arglist[nsa$v_arg_flag_alarm] = true; !  perform alarm audit
;  5083          5088  2
;  5084          5089  2  if .nsa$gr_journvec[nsa$v_evt_uaf]      ! If journal auditing,
;  5085          5090  2  then arglist[nsa$v_arg_flag_journ] = true; !  perform journal audit
;  5086          5091  2
;  5087          5092  2  if .pcb_sts[$bitposition(pcb$v_secaudit)] ! If mandatory auditing,
;  5088          5093  2  then arglist[nsa$v_arg_flag_mandy] = true; !  perform mandatory audit
;  5089          5094  2
;  5090          5095  2  if .arglist[nsa$b_arg_flag] eql 0       ! If no audit requested,
;  5091          5096  2  then return;                            !  simply exit
;  5092          5097  2
;  5093          5098  2  arglist[nsa$l_arg_id] = .code;          ! Set audit record type/subtype
;  5094          5099  2
;  5095          5100  2  arglist_ptr = arglist[nsa$t_arg_list]; ! Address packet(s) in arg list
;  5096          5101  2
;  5097          5102  2  if .arglist[nsa$w_arg_type] eql nsa$k_rectyp_sysuaf
;  5098          5103  2  then                                    ! For SYSUAF records...
;  5099          5104  3      begin
;  5100          5105  3      if .arglist[nsa$w_arg_subtype] neq nsa$k_rectyp_sysuaf_del
;  5101          5106  3      then
;  5102          5107  3          add_quad_packet(sysuaff, uaf$gq_sysuaff);
;  5103        P 5108  3      add_descr_packet(usernam,
;  5104        P 5109  3                       namelen(uaf$s_username, recbuf[uaf$t_username]),
;  5105          5110  3                       recbuf[uaf$t_username]);
;  5106          5111  3      if .arglist[nsa$w_arg_subtype] eql nsa$k_rectyp_sysuaf_cop
;  5107          5112  3      or .arglist[nsa$w_arg_subtype] eql nsa$k_rectyp_sysuaf_ren
;  5108          5113  3      then
;  5109        P 5114  3          add_descr_packet(usernam,
;  5110        P 5115  3                           namelen(uaf$s_username, .old_user),
;  5111          5116  3                           .old_user);
;  5112          5117  3      end
;  5113          5118  2  else                                    ! For NETUAF records...
;  5114          5119  3      begin
;  5115        P 5120  3      add_descr_packet(nodenam,
;  5116        P 5121  3                       namelen(naf$s_node, netbuf[naf$t_node]),
```

```
;  5117          5122  3                          netbuf[naf$t_node]);
;  5118        P 5123  3      add_descr_packet(usernam,
;  5119        P 5124  3                          namelen(naf$s_remuser, netbuf[naf$t_remuser]),
;  5120          5125  3                          netbuf[naf$t_remuser]);
;  5121        P 5126  3      add_descr_packet(usernam,
;  5122        P 5127  3                          namelen(naf$s_localuser, netbuf[naf$t_localuser]),
;  5123          5128  3                          netbuf[naf$t_localuser]);
;  5124          5129  3      if .arglist[nsa$w_arg_subtype] eql nsa$k_rectyp_netuaf_mod
;  5125          5130  3      then
;  5126          5131  4          begin
;  5127        P 5132  4          add_descr_packet(nodenam,
;  5128        P 5133  4                          namelen(naf$s_node, netbuf[naf$t_node]),
;  5129          5134  4                          netbuf[naf$t_node]);
;  5130        P 5135  4          add_descr_packet(usernam,
;  5131        P 5136  4                          namelen(naf$s_remuser, netbuf[naf$t_remuser]),
;  5132          5137  4                          netbuf[naf$t_remuser]);
;  5133        P 5138  4          add_descr_packet(usernam,
;  5134        P 5139  4                          namelen(uaf$s_username, .old_user),
;  5135          5140  4                          .old_user);
;  5136          5141  3          end;
;  5137          5142  2      end;
;  5138          5143  2
;  5139          5144  2 arglist[nsa$l_arg_count] = (.arglist_ptr - (arglist + 4)) / 4; ! Set # args
;  5140          5145  2
;  5141          5146  2 $cmkrnl(routin = nsa$event_audit, arglst = arglist); ! Do the security audit
;  5142          5147  2
;  5143          5148  1 end;


                                                .EXTRN  NSA$EVENT_AUDIT
                                                .EXTRN  NSA$GR_ALARMVEC
                                                .EXTRN  NSA$GR_JOURNVEC
                                                .EXTRN  SYS$CMKRNL

                                        007C 00000 SECURITY_AUDIT:
                                                .WORD   Save R2,R3,R4,R5,R6                     ;  5009
                          56 00000000'  00  9E 00002            MOVAB   NETBUF, R6
                          5E       AC   AE  9E 00009            MOVAB   -84(SP), SP
0054  8F        00        6E            00  2C 0000D            MOVC5   #0, (SP), #0, #84, ARGLIST      ;  5084
                                        6E      00014
                04 00000000G  00        02  E1 00015            BBC     #2, NSA$GR_ALARMVEC, 1$         ;  5086
                         08   AE        01  88 0001D            BISB2   #1, ARGLIST+8                   ;  5087
                04 00000000G  00        02  E1 00021 1$:        BBC     #2, NSA$GR_JOURNVEC, 2$         ;  5089
                         08   AE        02  88 00029            BISB2   #2, ARGLIST+8                   ;  5090
                04 00000000'  00        03  E1 0002D 2$:        BBC     #3, PCB_STS+3, 3$               ;  5092
                         08   AE        04  88 00035            BISB2   #4, ARGLIST+8                   ;  5093
                                  08    AE  95 00039 3$:        TSTB    ARGLIST+8                       ;  5095
                                        01  12 0003C            BNEQ    4$
                                        04  0003E               RET
                         04   AE       04   AC  D0 0003F 4$:    MOVL    CODE, ARGLIST+4                 ;  5098
                              52       0C   AE  9E 00044        MOVAB   ARGLIST+12, ARGLIST_PTR         ;  5100
                              02       04   AE  B1 00048        CMPW    ARGLIST+4, #2                   ;  5102
                                        3C  12  0004C           BNEQ    6$
                              02       06   AE  B1 0004E        CMPW    ARGLIST+6, #2                   ;  5105
                                        0F  13  00052           BEQL    5$
                              82 00030012   8F  D0 00054        MOVL    #196626, (ARGLIST_PTR)+         ;  5107
```

```
                                82      F9F8    C6  7D 0005B          MOVQ    UAF$GQ_SYSUAFF, (ARGLIST_PTR)+
                                        09      AE  96 00060          INCB    ARGLIST+9
                                82 0004000A     8F  D0 00063 5$:     MOVL    #262154, (ARGLIST_PTR)+                 5110
                FA80    C6      20              20  3A 0006A          LOCC    #32, #32, RECBUF+4
                        82      20              50  C3 00070          SUBL3   R0, #32, (ARGLIST_PTR)+
                                82      FA80    C6  9E 00074          MOVAB   RECBUF+4, (ARGLIST_PTR)+
                                        09      AE  96 00079          INCB    ARGLIST+9
                                04      06      AE  B1 0007C          CMPW    ARGLIST+6, #4                           5111
                                                7D  13 00080          BEQL    8$
                                05      06      AE  B1 00082          CMPW    ARGLIST+6, #5                           5112
                                                49  12 00086          BNEQ    7$
                                                75  11 00088          BRB     8$                                      5116
                                82 00040009     8F  D0 0008A 6$:     MOVL    #262153, (ARGLIST_PTR)+                 5122
                        66      20              20  3A 00091          LOCC    #32, #32, NETBUF
                        82      20              50  C3 00095          SUBL3   R0, #32, (ARGLIST_PTR)+
                                82              66  9E 00099          MOVAB   NETBUF, (ARGLIST_PTR)+
                                        09      AE  96 0009C          INCB    ARGLIST+9
                                82 0004000A     8F  D0 0009F          MOVL    #262154, (ARGLIST_PTR)+                 5125
                20      A6      20              20  3A 000A6          LOCC    #32, #32, NETBUF+32
                        82      20              50  C3 000AB          SUBL3   R0, #32, (ARGLIST_PTR)+
                                82      20      A6  9E 000AF          MOVAB   NETBUF+32, (ARGLIST_PTR)+
                                        09      AE  96 000B3          INCB    ARGLIST+9
                                82 0004000A     8F  D0 000B6          MOVL    #262154, (ARGLIST_PTR)+                 5128
                40      A6      20              20  3A 000BD          LOCC    #32, #32, NETBUF+64
                        82      20              50  C3 000C2          SUBL3   R0, #32, (ARGLIST_PTR)+
                                82      40      A6  9E 000C6          MOVAB   NETBUF+64, (ARGLIST_PTR)+
                                        09      AE  96 000CA          INCB    ARGLIST+9
                                03      06      AE  B1 000CD          CMPW    ARGLIST+6, #3                           5129
                                                43  12 000D1 7$:     BNEQ    9$
                                82 00040009     8F  D0 000D3          MOVL    #262153, (ARGLIST_PTR)+                 5134
                        66      20              20  3A 000DA          LOCC    #32, #32, NETBUF
                        82      20              50  C3 000DE          SUBL3   R0, #32, (ARGLIST_PTR)+
                                82              66  9E 000E2          MOVAB   NETBUF, (ARGLIST_PTR)+
                                        09      AE  96 000E5          INCB    ARGLIST+9
                                82 0004000A     8F  D0 000E8          MOVL    #262154, (ARGLIST_PTR)+                 5137
                20      A6      20              20  3A 000EF          LOCC    #32, #32, NETBUF+32
                        82      20              50  C3 000F4          SUBL3   R0, #32, (ARGLIST_PTR)+
                                82      20      A6  9E 000F8          MOVAB   NETBUF+32, (ARGLIST_PTR)+
                                        09      AE  96 000FC          INCB    ARGLIST+9
                                82 0004000A     8F  D0 000FF 8$:     MOVL    #262154, (ARGLIST_PTR)+                 5140
                08      BC      20              20  3A 00106          LOCC    #32, #32, @OLD_USER
                        82      20              50  C3 0010B          SUBL3   R0, #32, (ARGLIST_PTR)+
                                82      08      AC  D0 0010F          MOVL    OLD_USER, (ARGLIST_PTR)+
                                        09      AE  96 00113          INCB    ARGLIST+9
                                50      04      AE  9E 00116 9$:     MOVAB   ARGLIST+4, R0                           5147
                                52              50  C2 0011A          SUBL2   R0, R2
                        6E      52              04  C7 0011D          DIVL3   #4, R2, ARGLIST
                                                5E  DD 00121          PUSHL   SP                                      5146
                                    00000000G   00  9F 00123          PUSHAB  NSA$EVENT_AUDIT
                    00000000G   00              02  FB 00129          CALLS   #2, SYS$CMKRNL
                                                    04 00130          RET                                             5148

; Routine Size:  305 bytes,    Routine Base:  $CODE$ + 2A74
```

```
; 5145        5149  1 end
; 5146        5150  0 eludom                          ! End of module


                                                    .EXTRN  LIB$STOP


;                               PSECT SUMMARY

;       Name                    Bytes                   Attributes

;   $PLIT$                       2316  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $OWN$                        4364  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $GLOBAL$                     2032  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $CODE$                      11173  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   .  ABS  .                       0  NOVEC,NOWRT,NORD ,NOEXE,NOSHR,  LCL,  ABS,  CON,NOPIC,ALIGN(0)


;                           Library Statistics

;                                    --------  Symbols  --------      Pages       Processing
;       File                         Total     Loaded     Percent     Mapped      Time

;   _$255$DUA18:[SYSLIB]LIB.L32;1    18619        300           1       1000        00:01.4



;                                   COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:UAFMAIN/OBJ=OBJ$:UAFMAIN MSRC$:UAFMAIN/UPDATE=(BUG$:UAFMAIN)

; Size:          11173 code + 8712 data bytes
; Run Time:          02:11.0
; Elapsed Time:      05:57.8
; Lines/CPU Min:      2358
; Lexemes/CPU-Min:  38325
; Memory Used:   654 pages
; Compilation Complete
```

0450 AH-EF71A-SE
VAX/VMS V4.1 SRC LST MCRF UPD

UAFMAIN
LIS

UAF

AUTHORIZE
MAP

0451 AH-EF71A-SE
VAX/VMS V4.1 SRC LST MCRF UPD

UAFPARSE
LIS

BOOTDRUVIP
LIS

VMBUVAX.P
LIS

UVI.ROM

VMBUVAXIP
MAP

CONIO
LIS

UVMS

REMOVE
COM