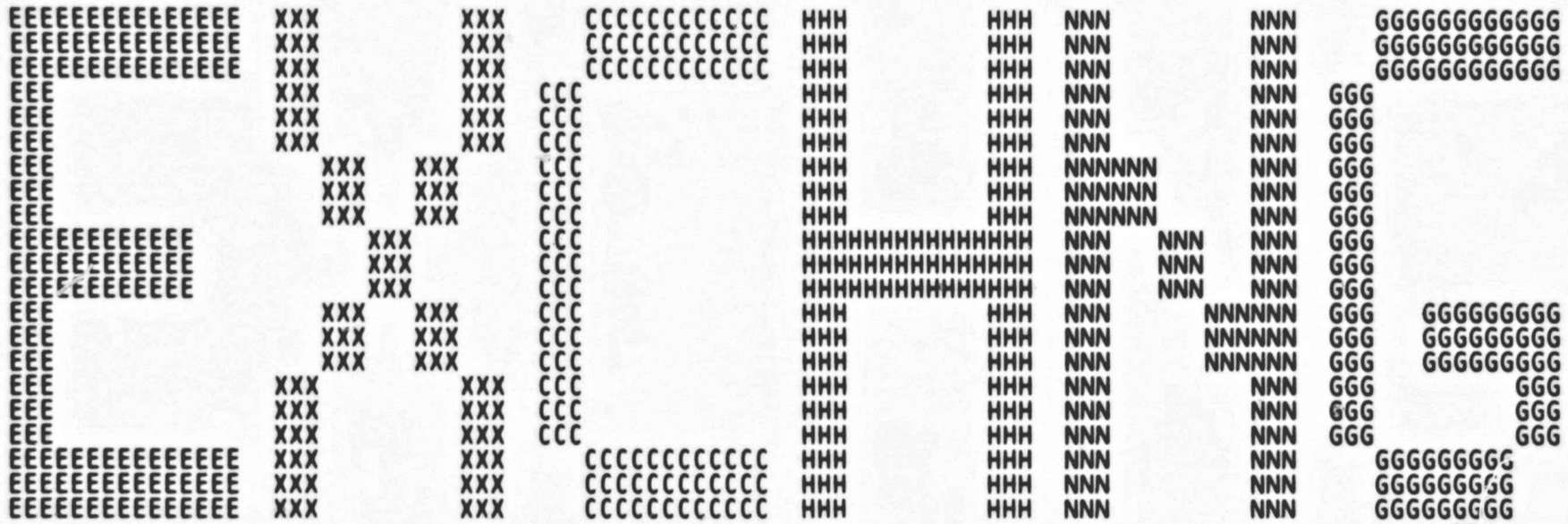
EEEEEEEEEEEEEEE XXX XXX CCCCCCCCCCCC HHH HHH NNN NNN GGGGGGGGGGG
EEEEEEEEEEEEEEE XXX XXX CCCCCCCCCCCC HHH HHH NNN NNN GGGGGGGGGGG
EEEEEEEEEEEEEEE XXX XXX CCCCCCCCCCCC HHH HHH NNN NNN GGGGGGGGGGG
EEE XXX XXX CCC HHH HHH NNN NNN GGG
EEE XXX XXX CCC HHH HHH NNN NNN GGG
EEE XXX XXX CCC HHH HHH NNN NNN GGG
EEE XXX XXX CCC HHH HHH NNNNNN NNN GGG
EEE XXX XXX CCC HHH HHH NNNNNN NNN GGG
EEE XXX XXX CCC HHH HHH NNNNNN NNN GGG
EEEEEEEEEEEE XXX CCC HHHHHHHHHHHHHHH NNN NNN NNN GGG
EEEEEEEEEEEE XXX CCC HHHHHHHHHHHHHHH NNN NNN NNN GGG
EEEEEEEEEEEE XXX CCC HHHHHHHHHHHHHHH NNN NNN NNN GGG
EEE XXX XXX CCC HHH HHH NNN NNNNNN GGG GGGGGGGG
EEE XXX XXX CCC HHH HHH NNN NNNNNN GGG GGGGGGGG
EEE XXX XXX CCC HHH HHH NNN NNNNNN GGG GGGGGGGG
EEE XXX XXX CCC HHH HHH NNN NNN GGG GGG
EEE XXX XXX CCC HHH HHH NNN NNN GGG GGG
EEE XXX XXX CCC HHH HHH NNN NNN GGG GGG
EEEEEEEEEEEEEEE XXX XXX CCCCCCCCCCCC HHH HHH NNN NNN GGGGGGGGG
EEEEEEEEEEEEEEE XXX XXX CCCCCCCCCCCC HHH HHH NNN NNN GGGGGGGGG
EEEEEEEEEEEEEEE XXX XXX CCCCCCCCCCCC HHH HHH NNN NNN GGGGGGGGG

```
EEEEEEEEEE  XX      XX    CCCCCCCC  MM      MM    AAAAAA     IIIIII    NN      NN
EEEEEEEEEE  XX      XX    CCCCCCCC  MM      MM    AAAAAA     IIIIII    NN      NN
EE          XX      XX  CC          MMMM  MMMM  AA      AA     II     NN      NN
EE          XX      XX  CC          MMMM  MMMM  AA      AA     II     NN      NN
EE            XX  XX    CC          MM  MM  MM  AA      AA     II     NNNN    NN
EE            XX  XX    CC          MM  MM  MM  AA      AA     II     NNNN    NN
EEEEEEEE        XX      CC          MM      MM  AA      AA     II     NN  NN  NN
EEEEEEEE        XX      CC          MM      MM  AA      AA     II     NN  NN  NN
EE            XX  XX    CC          MM      MM  AAAAAAAAAA     II     NN    NNNN
EE            XX  XX    CC          MM      MM  AAAAAAAAAA     II     NN    NNNN
EE          XX      XX  CC          MM      MM  AA      AA     II     NN      NN    ....
EE          XX      XX  CC          MM      MM  AA      AA     II     NN      NN    ....
EEEEEEEEEE  XX      XX    CCCCCCCC  MM      MM  AA      AA  IIIIII    NN      NN    ....
EEEEEEEEEE  XX      XX    CCCCCCCC  MM      MM  AA      AA  IIIIII    NN      NN    ....

LL            IIIIII      SSSSSSSS
LL            IIIIII    SSSSSSSSS
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  exch$main                                %TITLE 'Image transfer point, command dispatcher'
;     2          0002  0                 (
;     3          0003  0                 IDENT = 'V04-000',
;     4          0004  0                 ADDRESSING_MODE (EXTERNAL=LONG_RELATIVE, NONEXTERNAL=WORD_RELATIVE),
;     5          0005  0                 MAIN = main_start
;     6          0006  0                 ) =
;     7          0007  1 BEGIN
;     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:     EXCHANGE - Foreign volume interchange utility
;    34          0034  1 !
;    35          0035  1 ! ABSTRACT:     This program supports foreign volumes for VAX/VMS.  A subset  of the DCL command language is
;    36          0036  1 !               supported to perform operations as requested on the foreign volume.  The subset DCL commands
;    37          0037  1 !               are:
;    38          0038  1 !                       COPY       - transfer a file
;    39          0039  1 !                       DELETE     - delete a file
;    40          0040  1 !                       DIRECTORY  - list files on volume
;    41          0041  1 !                       DISMOUNT   - dismount a mounted volume
;    42          0042  1 !                       EXIT       - exit program
;    43          0043  1 !                       HELP       - ask for explanation
;    44          0044  1 !                       INITIALIZE - create empty volume
;    45          0045  1 !                       MOUNT      - mount a foreign volume
;    46          0046  1 !                       RENAME     - rename a file
;    47          0047  1 !                       SQUEEZE    - compress a volume
;    48          0048  1 !                       TYPE       - display a file on SYS$OUTPUT
;    49          0049  1 !
;    50          0050  1 ! ENVIRONMENT:  VAX/VMS operating system, unprivileged user-mode utility
;    51          0051  1 !
;    52          0052  1 ! AUTHOR:       CW Hobbs        CREATION DATE: 1-July-1982
;    53          0053  1 !
;    54          0054  1 ! MODIFIED BY:
;    55          0055  1 !
;    56          0056  1 !       V03-002 CWH3002                 CW Hobbs                12-Apr-1984
;    57          0057  1 !               Add conditional to prevent noise messages during debugging.
```

```
;    58          0058  1 !
;    59          0059  1 !
;    60          0060  1 !--
;    61          0061  1
;    62          0062  1 ! Include files:
;    63          0063  1 !
;    64          0064  1 MACRO $module_name_string = 'exch$main' %;      ! The require file needs to know our module name
;    65          0065  1 REQUIRE 'SRC$:EXCREQ'                           ! Facility-wide require file
;    66          0066  1     ;
```

```
;    68          0163  1 %SBTTL 'Module table of contents'
;    69          0164  1
;    70          0165  1 ! Module table of contents:
;    71          0166  1 !
;    72          0167  1 FORWARD ROUTINE
;    73          0168  1     main_control_c_ast     : NOVALUE,           ! AST routine to set control/c flag
;    74          0169  1     exch$main_exit,                             ! EXIT verb dispatch routine
;    75          0170  1     main_exit_handler      : NOVALUE,           ! Dismount volumes during exit
;    76          0171  1     main_handle_cli_nocomd,                     ! Condition handler for CLI$_NOCOMD error
;    77          0172  1     exch$main_help,                             ! HELP verb dispatch routine
;    78          0173  1     main_setup_create_excg : NOVALUE,           ! Allocate and initialize the global data
;    79          0174  1     main_setup_load_time   : NOVALUE,           ! Once-only load time initializations
;    80          0175  1     main_start                                  ! Transfer point, outer command loop
;    81          0176  1     ;
;    82          0177  1
;    83          0178  1 ! System library routines
;    84          0179  1 !
;    85          0180  1 EXTERNAL ROUTINE
;    86          0181  1     lbr$output_help : ADDRESSING_MODE(GENERAL)  ! Librarian get help
;    87          0182  1     ;
;    88          0183  1
;    89          0184  1 ! EXCHANGE facility routines
;    90          0185  1 !
;    91          0186  1 EXTERNAL ROUTINE
;    92          0187  1     exch$moun_dismount_action,                  ! Dismount mounted volumes
;    93          0188  1     exch$util_file_error,                       ! Signal RMS error
;    94          0189  1     exch$util_vm_allocate_zeroed                ! Allocate virtual memory
;    95          0190  1     ;
;    96          0191  1
;    97          0192  1 ! Read-only GLOBAL storage
;    98          0193  1 !
;    99          0194  1 GLOBAL
;   100          0195  1     exch$gq_dyn_str_template : $dyn_str_desc    ! An initialized, null dynamic string descriptor
;   101          0196  1     ;
;   102          0197  1
;   103          0198  1 ! Read-write GLOBAL storage
;   104          0199  1 !
;   105          0200  1 $global_rw
;   106          0201  1     exch$a_gbl  : REF BLOCK [,BYTE]             ! The pointer to everything else
;   107          0202  1     ;
;   108          0203  1
;   109          0204  1 ! Bound declarations:
;   110          0205  1 !
;   111          0206  1 !BIND
;   112          0207  1 !    ;
;   113          0208  1
;   114          0209  1 ! Local macros
;   115          0210  1 !
;   116          0211  1 MACRO
;   117        M 0212  1     $$offset_check (sym) =      ! Check that context block offsets coincide
;   118        M 0213  1                                 (($BYTEOFFSET (%NAME ('ctx$',sym)) EQL $BYTEOFFSET (%NAME ('dos11ctx
;   119        M 0214  1                                   AND
;   120        M 0215  1                                  ($BYTEOFFSET (%NAME ('ctx$',sym)) EQL $BYTEOFFSET (%NAME ('rt11ctx$
;   121          0216  1                                 %,
;   122          0217  1
;   123        M 0218  1     $$bit_check (sym) =         ! Check that context block bitfields coincide
;   124        M 0219  1                                 (($BYTEOFFSET (%NAME ('ctx$',sym)) EQL $BYTEOFFSET (%NAME ('dos11ctx
```

```
  AND
 ($BITPOSITION (%NAME ('ctx$',sym)) EQL $BITPOSITION (%NAME ('dos11c
  AND
 ($BYTEOFFSET (%NAME ('ctx$',sym)) EQL $BYTEOFFSET (%NAME ('rt11ctx$
  AND
 ($BITPOSITION (%NAME ('ctx$',sym)) EQL $BITPOSITION (%NAME ('rt11ct
%;
```

```
;   133         0227  1 GLOBAL ROUTINE main_control_c_ast : NOVALUE =   %SBTTL 'main_control_c_ast'
;   134         0228  2 BEGIN
;   135         0229  2 !++
;   136         0230  2 !
;   137         0231  2 ! FUNCTIONAL DESCRIPTION:
;   138         0232  2 !
;   139         0233  2 !       Set a flag which says that a control/c ast has been received
;   140         0234  2 !
;   141         0235  2 ! INPUTS:
;   142         0236  2 !
;   143         0237  2 !       none
;   144         0238  2 !
;   145         0239  2 ! IMPLICIT INPUTS:
;   146         0240  2 !
;   147         0241  2 !       none
;   148         0242  2 !
;   149         0243  2 ! OUTPUTS:
;   150         0244  2 !
;   151         0245  2 !       none
;   152         0246  2 !
;   153         0247  2 ! IMPLICIT OUTPUTS:
;   154         0248  2 !
;   155         0249  2 !       exch$a_gbl [excg$v_control_c] - flag set that we have received the ast
;   156         0250  2 !
;   157         0251  2 ! ROUTINE VALUE:
;   158         0252  2 !
;   159         0253  2 !       none
;   160         0254  2 !
;   161         0255  2 ! SIDE EFFECTS:
;   162         0256  2 !
;   163         0257  2 !       none
;   164         0258  2 !--
;   165         0259  2
;   166         0260  2 $dbgtrc_prefix ('main_control_c_ast> ');
;   167         0261  2 $trace_print_lit ('received control/c ast');
;   168         0262  2
;   169         0263  2 ! Set the bit which says that an AST has been delivered
;   170         0264  2 !
;   171         0265  2 exch$a_gbl [excg$v_control_c] = true;
;   172         0266  2
;   173         0267  2 RETURN;
;   174         0268  1 END;


                                                .TITLE  EXCH$MAIN Image transfer point, command dispatc
                                                        her
;                                               .IDENT  \V04-000\

                                                .PSECT  EXCH$RW_GLOBAL,NOEXE,2

                                    00000 EXCH$A_GBL::
                                                .BLKB   4

                                                .PSECT  EXCH$MAIN_PLIT,NOWRT,2

                                0000  00000 EXCH$GQ_DYN_STR_TEMPLATE::
                                                .WORD   0                                       ;
```

```
                                02  0E  00002          .BYTE   14, 2                                   ;
                          00000000  00004              .LONG   0                                       ;

                                                       .EXTRN  LBR$OUTPUT_HELP
                                                       .EXTRN  EXCH$MOUN_DISMOUNT_ACTION
                                                       .EXTRN  EXCH$UTIL_FILE_ERROR
                                                       .EXTRN  EXCH$UTIL_VM_ALLOCATE_ZEROED

                                                       .PSECT  EXCH$MAIN_CODE,NOWRT,2

                                      0000 00000       .ENTRY  MAIN_CONTROL_C_AST, Save nothing        ; 0227
                    00000000'  FF       01  88 00002  BISB2   #1, @EXCH$A_GBL                         ; 0265
                                           04 00009  RET                                             ; 0268

; Routine Size:  10 bytes,    Routine Base:  EXCH$MAIN_CODE + 0000
```

```
;  176          0269  1 GLOBAL ROUTINE exch$main_exit (error_code) =    %SBTTL 'exch$main_exit (error_code)'
;  177          0270  2 BEGIN
;  178          0271  2 !++
;  179          0272  2 !
;  180          0273  2 ! FUNCTIONAL DESCRIPTION:
;  181          0274  2 !
;  182          0275  2 !       Action routine for the EXIT verb, parses and performs main control functions for EXIT
;  183          0276  2 !
;  184          0277  2 ! INPUTS:
;  185          0278  2 !
;  186          0279  2 !       error_code - final status code, return ss$_normal if it is 0
;  187          0280  2 !
;  188          0281  2 ! IMPLICIT INPUTS:
;  189          0282  2 !
;  190          0283  2 !       Command parameters and qualifiers as returned from CLI$xxx routines.
;  191          0284  2 !
;  192          0285  2 ! OUTPUTS:
;  193          0286  2 !
;  194          0287  2 !       none
;  195          0288  2 !
;  196          0289  2 ! IMPLICIT OUTPUTS:
;  197          0290  2 !
;  198          0291  2 !       none
;  199          0292  2 !
;  200          0293  2 ! ROUTINE VALUE:
;  201          0294  2 !
;  202          0295  2 !       none
;  203          0296  2 !
;  204          0297  2 ! SIDE EFFECTS:
;  205          0298  2 !
;  206          0299  2 !       EXCHANGE will exit to the DCL command level.
;  207          0300  2 !--
;  208          0301  2
;  209          0302  2 $dbgtrc_prefix ('main_exit> ');
;  210          0303  2
;  211          0304  2 $debug_print_lit ('EXIT verb');
;  212          0305  2
;  213          0306  2 IF .error_code EQL 0
;  214          0307  2 THEN
;  215          0308  3     $exit (code=ss$_normal)
;  216          0309  2 ELSE
;  217          0310  2     $exit (code=.error_code);
;  218          0311  2
;  219          0312  2 RETURN 0;
;  220          0313  1 END;


                                                        .EXTRN  SYS$EXIT

                                          0000 00000    .ENTRY  EXCH$MAIN_EXIT, Save nothing                    ; 0269
                                    04    AC   D5 00002         TSTL    ERROR_CODE                              ; 0306
                                          04   12 00005         BNEQ    1$                                      ;
                                          01   DD 00007         PUSHL   #1                                      ; 0308
                                          03   11 00009         BRB     2$                                      ;
                                    04    AC   DD 0000B 1$:     PUSHL   ERROR_CODE                              ; 0310
                00000000G   00            01   FB 0000E 2$:     CALLS   #1, SYS$EXIT                            ;
```

```
                                50  D4 00015          CLRL    R0                                      ; 0312
                                    04 00017          RET                                             ; 0313

; Routine Size:  24 bytes,    Routine Base:  EXCH$MAIN_CODE + 000A
```

```
;   222          0314  1 GLOBAL ROUTINE main_exit_handler (status) : NOVALUE =   %SBTTL 'main_exit_handler (status)'
;   223          0315  2 BEGIN
;   224          0316  2 !++
;   225          0317  2 !
;   226          0318  2 ! FUNCTIONAL DESCRIPTION:
;   227          0319  2 !
;   228          0320  2 !       Perform exit functions for EXCHANGE.  Dismount mounted volumes.  Currently only necessary for RT-11
;   229          0321  2 !       volumes with global caching active.
;   230          0322  2 !
;   231          0323  2 ! INPUTS:
;   232          0324  2 !
;   233          0325  2 !       status - reason for exit
;   234          0326  2 !
;   235          0327  2 ! IMPLICIT INPUTS:
;   236          0328  2 !
;   237          0329  2 !       mounted volume queue
;   238          0330  2 !
;   239          0331  2 ! OUTPUTS:
;   240          0332  2 !
;   241          0333  2 !       none
;   242          0334  2 !
;   243          0335  2 ! IMPLICIT OUTPUTS:
;   244          0336  2 !
;   245          0337  2 !       none
;   246          0338  2 !
;   247          0339  2 ! ROUTINE VALUE:
;   248          0340  2 !
;   249          0341  2 !       none
;   250          0342  2 !
;   251          0343  2 ! SIDE EFFECTS:
;   252          0344  2 !
;   253          0345  2 !       lots
;   254          0346  2 !--
;   255          0347  2
;   256          0348  2 $dbgtrc_prefix ('main_exit_handler> ');
;   257          0349  2
;   258          0350  2 LOCAL
;   259          0351  2     ptr : $ref_bblock
;   260          0352  2     ;
;   261          0353  2
;   262          0354  2 $trace_print_lit ('entering exit handler');
```

```
;   264         0355  2 ! Set the flag that we are exiting
;   265         0356  2 !
;   266         0357  2 exch$a_gbl [excg$v_exiting] = true;
;   267         0358  2
;   268         0359  2 ! Loop through the queue of in-use volbs.  We must go to the head of the queue with each loop, since the
;   269         0360  2 ! dismount routine will remove the current item from the queue.
;   270         0361  2 !
;   271         0362  2 WHILE 1
;   272         0363  2 DO
;   273         0364  3     BEGIN
;   274         0365  3     ptr = .exch$a_gbl [excg$a_volb_use_flink];          ! Get the first mounted volb in the queue
;   275         0366  3     IF .ptr EQL exch$a_gbl [excg$a_volb_use_flink]      ! If same as header, we are done
;   276         0367  3     THEN
;   277         0368  3         EXITLOOP;
;   278         0369  3     $block_check (2, .ptr, volb, 637);                  ! Make sure it is a volb
;   279         0370  3
;   280         0371  3     ! If there are any modified segments, and the device is slow, tell them we are flushing
;   281         0372  3     !
;   282         0373  3     IF .ptr [volb$l_dircache] NEQ volb$m_dircache_active
;   283         0374  3     THEN
;   284         0375  3         IF .ptr [volb$l_devtype] EQL dt$_tu58   ! If it is any kind of TU58
;   285         0376  3         THEN
;   286         0377  4             BEGIN
;   287         0378  4             LOCAL
;   288         0379  4                 msgvec : VECTOR [5, LONG],
;   289         0380  4                 status;
;   290         0381  4
;   291         0382  4             ! We use the $putmsg service to print this message.  If we signalled it, we could exit the image
;   292         0383  4             ! another signal was active in the catch-all condition handler.  This is extremely likely to hap
;   293         0384  4             ! if the control/Y was hit during a command with a /LOG in effect, since the catch-all handler e
;   294         0385  4             ! up printing EXCHANGE log messages.
;   295         0386  4             !
;   296         0387  4             msgvec [0] = 4;
;   297         0388  4             msgvec [1] = exch$_writecache;
;   298         0389  4             msgvec [2] = 2;
;   299         0390  4             msgvec [3] = .ptr [volb$l_vol_ident_len];
;   300         0391  4             msgvec [4] = ptr [volb$t_vol_ident];
;   301         0392  5             IF NOT (status = $putmsg (msgvec=msgvec))
;   302         0393  4             THEN
;   303         0394  4                 $exch_signal_stop (.status);
;   304         0395  3             END;
;   305         0396  3
;   306         0397  3     ! Now call the action routine, so that in effect we will do a standard DISMOUNT of the volume
;   307         0398  3     !
;   308         0399  3     exch$moun_dismount_action (.ptr);
;   309         0400  2     END;
;   310         0401  2
;   311         0402  2 RETURN;
;   312         0403  1 END;


                                                    .EXTRN  EXCH$UTIL_BLOCK_CHECK
                                                    .EXTRN  EXCH$_WRITECACHE
                                                    .EXTRN  SYS$PUTMSG, LIB$STOP

                                    000C 00000      .ENTRY  MAIN_EXIT_HANDLER, Save R2,R3                   ; 0314
```

```
                        5E            14  C2 00002        SUBL2   #20, SP                                ;
             00000000'  FF            10  88 00005        BISB2   #16, @EXCH$A_GBL                       ; 0357
                        50 00000000'  EF  D0 0000C 1$:    MOVL    EXCH$A_GBL, R0                         ; 0365
                        53     00C0   C0  D0 00013        MOVL    192(R0), PTR                           ;
                        50     00C0   C0  9E 00018        MOVAB   192(R0), R0                            ; 0366
                        50            53  D1 0001D        CMPL    PTR, R0                                ;
                                      60  13 00020        BEQL    3$                                     ;
                        52 041B00F3   8F  D0 00022        MOVL    #68878579, R2                          ; 0369
                        51     027D   8F  3C 00029        MOVZWL  #637, R1                               ;
                        50            53  D0 0002E        MOVL    PTR, R0                                ;
                           00000000G  EF  16 00031        JSB     EXCH$UTIL_BLOCK_CHECK                  ;
                        01       50   A3  D1 00037        CMPL    80(PTR), #1                            ; 0373
                                      3A  13 0003B        BEQL    2$                                     ;
                        0E       3C   A3  D1 0003D        CMPL    60(PTR), #14                           ; 0375
                                      34  12 00041        BNEQ    2$                                     ;
                        6E            04  D0 00043        MOVL    #4, MSGVEC                             ; 0387
                 04     AE 00000000G  8F  D0 00046        MOVL    #EXCH$_WRITECACHE, MSGVEC+4            ; 0388
                 08     AE            02  D0 0004E        MOVL    #2, MSGVEC+8                           ; 0389
                 0C     AE       65   A3  D0 00052        MOVL    101(PTR), MSGVEC+12                    ; 0390
                 10     AE       69   A3  9E 00057        MOVAB   105(R3), MSGVEC+16                     ; 0391
                                      7E  7C 0005C        CLRQ    -(SP)                                  ; 0392
                                      7E  D4 0005E        CLRL    -(SP)                                  ;
                                 0C   AE  9F 00060        PUSHAB  MSGVEC                                 ;
             00000000G  00            04  FB 00063        CALLS   #4, SYS$PUTMSG                         ;
                        0A            50  E8 0006A        BLBS    STATUS, 2$                             ;
                                      50  DD 0006D        PUSHL   STATUS                                 ; 0394
             00000000G  00            01  FB 0006F        CALLS   #1, LIB$STOP                           ;
                                          04 00076        RET                                            ;
                                      53  DD 00077 2$:    PUSHL   PTR                                    ; 0399
             00000000G  EF            01  FB 00079        CALLS   #1, EXCH$MOUN_DISMOUNT_ACTION          ;
                                      8A  11 00080        BRB     1$                                     ; 0362
                                          04 00082 3$:    RET                                            ; 0403

; Routine Size:  131 bytes,    Routine Base:  EXCH$MAIN_CODE + 0022
```

```
;  314     0404  1 GLOBAL ROUTINE main_handle_cli_nocomd (sig : $ref_bblock, mech : $ref_bblock) = %SBTTL 'main_handle_cli_noco
;  315     0405  2 BEGIN
;  316     0406  2 !++
;  317     0407  2 !
;  318     0408  2 ! FUNCTIONAL DESCRIPTION:
;  319     0409  2 !
;  320     0410  2 !       This routine intercepts the signal MSG$_COMD.  This is used to avoid unnecessary noise when
;  321     0411  2 !       a blank line is given.
;  322     0412  2 !
;  323     0413  2 ! INPUTS:
;  324     0414  2 !
;  325     0415  2 !       sig  - signal argument list
;  326     0416  2 !       mech - mechanism argument list
;  327     0417  2 !
;  328     0418  2 ! IMPLICIT INPUTS:
;  329     0419  2 !
;  330     0420  2 !       none
;  331     0421  2 !
;  332     0422  2 ! OUTPUTS:
;  333     0423  2 !
;  334     0424  2 !       none
;  335     0425  2 !
;  336     0426  2 ! IMPLICIT OUTPUTS:
;  337     0427  2 !
;  338     0428  2 !       none
;  339     0429  2 !
;  340     0430  2 ! ROUTINE VALUE:
;  341     0431  2 !
;  342     0432  2 !       SS$_CONTINUE if the signal was CLI$_NOCOMD, otherwise SS$_RESIGNAL.
;  343     0433  2 !
;  344     0434  2 ! SIDE EFFECTS:
;  345     0435  2 !
;  346     0436  2 !       error message is not printed for nocomd errors
;  347     0437  2 !--
;  348     0438  2
;  349     0439  2 $dbgtrc_prefix ('main_handle_cli_nocomd> ');
;  350     0440  2
;  351     0441  2 ! If the signal name is what we are looking for, then do interrupt the signal
;  352     0442  2 !
;  353     0443  2 IF .sig [chf$l_sig_name] EQL cli$_nocomd        ! DCL CLI error message (sinful knowlege of what DCL does!)
;  354     0444  2 THEN
;  355     0445  2     RETURN ss$_continue;
;  356     0446  2
;  357     0447  2 RETURN ss$_resignal;
;  358     0448  1 END;


                                                        .EXTRN  CLI$_NOCOMD

                                        0000 00000      .ENTRY  MAIN_HANDLE_CLI_NOCOMD, Save nothing    ; 0404
                    50      04  AC    D0 00002        MOVL    SIG, R0                                 ; 0443
        00000000G   8F      04  A0    D1 00006        CMPL    4(R0), #CLI$_NOCOMD                     ;
                                04    12 0000E        BNEQ    1$                                      ;
                    50              01  D0 00010      MOVL    #1, R0                                  ; 0445
                                        04 00013      RET                                             ;
                    50      0918    8F  3C 00014 1$:  MOVZWL  #2328, R0                               ; 0447
```

```
                                        04 00019        RET                                     ; 0448

; Routine Size:  26 bytes,    Routine Base:  EXCH$MAIN_CODE + 00A5
```

```
;  360     0449  1 GLOBAL ROUTINE exch$main_help = %SBTTL 'exch$main_help)'
;  361     0450  2 BEGIN
;  362     0451  2 !++
;  363     0452  2 !
;  364     0453  2 ! FUNCTIONAL DESCRIPTION:
;  365     0454  2 !
;  366     0455  2 !       Action routine for the HELP verb, parses and performs main control functions for HELP
;  367     0456  2 !
;  368     0457  2 ! INPUTS:
;  369     0458  2 !
;  370     0459  2 !       none
;  371     0460  2 !
;  372     0461  2 ! IMPLICIT INPUTS:
;  373     0462  2 !
;  374     0463  2 !       Command parameters and qualifiers as returned from CLI$xxx routines.
;  375     0464  2 !
;  376     0465  2 ! OUTPUTS:
;  377     0466  2 !
;  378     0467  2 !       none
;  379     0468  2 !
;  380     0469  2 ! IMPLICIT OUTPUTS:
;  381     0470  2 !
;  382     0471  2 !       none
;  383     0472  2 !
;  384     0473  2 ! ROUTINE VALUE:
;  385     0474  2 !
;  386     0475  2 !       none
;  387     0476  2 !
;  388     0477  2 ! SIDE EFFECTS:
;  389     0478  2 !
;  390     0479  2 !       Help Librarian routines will be entered.
;  391     0480  2 !--
;  392     0481  2
;  393     0482  2 $dbgtrc_prefix ('main_help> ');
;  394     0483  2
;  395     0484  2 LOCAL
;  396     0485  2     status,
;  397     0486  2     topic : $desc_block
;  398     0487  2     ;
;  399     0488  2
;  400     0489  2 $debug_print_lit ('HELP verb');
;  401     0490  2
;  402     0491  2 $dyn_str_desc_init (topic);
;  403     0492  2
;  404     0493  2 cli$get_value (%ASCID 'TOPIC', topic);
;  405     0494  2
;  406     0495  2 status = lbr$output_help
;  407     0496  2                  (lib$put_output,
;  408     0497  2                  0,
;  409     0498  2                  topic,
;  410     0499  2                  %ASCID 'EXCHNGHLP',
;  411     0500  2                  %REF (HLP$M_PROMPT OR HLP$M_PROCESS OR HLP$M_GROUP OR HLP$M_SYSTEM OR HLP$M_HELP),
;  412     0501  2                  lib$get_input);
;  413     0502  2
;  414     0503  2 IF NOT .status
;  415     0504  2 THEN
;  416     0505  2     $exch_signal ($warning_stat_copy (.status));
```

```
;  417          0506  2
;  418          0507  2 RETURN .status;
;  419          0508  1 END;


                                                                .PSECT  EXCH$MAIN_PLIT,NOWRT,2

                        00  00  00  43  49  50  4F  54  00008 P.AAB:   .ASCII  \TOPIC\<0><0><0>                                ;
                                                010E0005  00010 P.AAA:   .LONG   17694725                                      ;
                                                00000000' 00014          .ADDRESS P.AAB                                       ;
                00  00  00  50  4C  48  47  4E  48  43  58  45  00018 P.AAD:   .ASCII  \EXCHNGHLP\<0><0><0>                     ;
                                                010E0009  00024 P.AAC:   .LONG   17694729                                      ;
                                                00000000' 00028          .ADDRESS P.AAD                                       ;

                                                          TMPL=                 EXCH$GQ_DYN_STR_TEMPLATE
                                                                .EXTRN  CLI$GET_VALUE, LIB$PUT_OUTPUT
                                                                .EXTRN  LIB$GET_INPUT

                                                                .PSECT  EXCH$MAIN_CODE,NOWRT,2

                                                    0004 00000          .ENTRY  EXCH$MAIN_HELP, Save R2                      ; 0449
                                                5E      0C  C2 00002    SUBL2   #12, SP                                      ;
                                          04    AE  0000' CF  7D 00005    MOVQ    TMPL, DESC                                   ; 0491
                                                04  AE  9F 0000B         PUSHAB  TOPIC                                        ; 0493
                                              0000' CF  9F 0000E         PUSHAB  P.AAA                                        ;
                                00000000G     00  02  FB 00012          CALLS   #2, CLI$GET_VALUE                            ;
                                          00000000G 00  9F 00019         PUSHAB  LIB$GET_INPUT                                ; 0496
                                          04    AE  2F  D0 0001F         MOVL    #47, 4(SP)                                   ; 0500
                                                04  AE  9F 00023         PUSHAB  4(SP)                                        ;
                                              0000' CF  9F 00026         PUSHAB  P.AAC                                        ; 0498
                                                10  AE  9F 0002A         PUSHAB  TOPIC                                        ; 0496
                                                    7E  D4 0002D         CLRL    -(SP)                                        ;
                                          00000000G 00  9F 0002F         PUSHAB  LIB$PUT_OUTPUT                               ;
                                00000000G     00  06  FB 00035          CALLS   #6, LBR$OUTPUT_HELP                          ;
                                              52  50  D0 0003C         MOVL    R0, STATUS                                   ;
                                              0F  52  E8 0003F         BLBS    STATUS, 1$                                   ; 0503
                                              50  52  D0 00042         MOVL    STATUS, STATUS2                              ; 0505
                                              50  07  8A 00045         BICB2   #7, STATUS2                                  ;
                                                  50  DD 00048         PUSHL   STATUS2                                      ;
                                00000000G     00  01  FB 0004A          CALLS   #1, LIB$SIGNAL                               ;
                                              50  52  D0 00051 1$:      MOVL    STATUS, R0                                   ; 0507
                                                  04  00054          RET                                                  ; 0508

; Routine Size:  85 bytes,    Routine Base:  EXCH$MAIN_CODE + 00BF
```

```
;  421     0509  1 GLOBAL ROUTINE main_setup_create_excg : NOVALUE =       %SBTTL 'main_setup_create_excg'
;  422     0510  2 BEGIN
;  423     0511  2 !++
;  424     0512  2 !
;  425     0513  2 ! FUNCTIONAL DESCRIPTION:
;  426     0514  2 !
;  427     0515  2 !       This routine allocates and initializes the global data
;  428     0516  2 !
;  429     0517  2 ! INPUTS:
;  430     0518  2 !
;  431     0519  2 !       none
;  432     0520  2 !
;  433     0521  2 ! IMPLICIT INPUTS:
;  434     0522  2 !
;  435     0523  2 !       none
;  436     0524  2 !
;  437     0525  2 ! OUTPUTS:
;  438     0526  2 !
;  439     0527  2 !       none
;  440     0528  2 !
;  441     0529  2 ! IMPLICIT OUTPUTS:
;  442     0530  2 !
;  443     0531  2 !       exch$a_gbl - external pointer to the block
;  444     0532  2 !
;  445     0533  2 ! ROUTINE VALUE:
;  446     0534  2 !
;  447     0535  2 !       none
;  448     0536  2 !
;  449     0537  2 ! SIDE EFFECTS:
;  450     0538  2 !
;  451     0539  2 !       Memory is allocated
;  452     0540  2 !--
;  453     0541  2
;  454     0542  2 $dbgtrc_prefix ('main_setup_create_excg> ');
;  455     0543  2
;  456     0544  2 LOCAL
;  457     0545  2     ptr
;  458     0546  2     ;
;  459     0547  2
;  460     0548  2
;  461     0549  2 $debug_print_lit ('entry');
;  462     0550  2
;  463     0551  2 ! Allocate the global data structure
;  464     0552  2 !
;  465     0553  2 exch$a_gbl = exch$util_vm_allocate_zeroed (exchblk$s_excg);
;  466     0554  2
;  467     0555  2 ! Set the block size and type
;  468     0556  2 !
;  469     0557  2 $block_init( .exch$a_gbl, excg);
;  470     0558  2
;  471     0559  2 ! Init the queue headers for the global resources
;  472     0560  2 !
;  473     0561  2 $queue_initialize (exch$a_gbl [excg$q_dos11ctx_use]);   ! Head of queue of all $DOS11CTX's in use
;  474     0562  2 $queue_initialize (exch$a_gbl [excg$q_dos11ctx_avl]);   ! Head of queue of all available $DOS11CTX's
;  475     0563  2
;  476     0564  2 $queue_initialize (exch$a_gbl [excg$q_filb_use]);       ! Head of queue of all $FILB's in use
;  477     0565  2 $queue_initialize (exch$a_gbl [excg$q_filb_avl]);       ! Head of queue of all available $FILB's
```

```
;  478          0566  2
;  479          0567  2 $queue_initialize (exch$a_gbl [excg$q_namb_use]);       ! Head of queue of all $NAMB's in use
;  480          0568  2 $queue_initialize (exch$a_gbl [excg$q_namb_avl]);       ! Head of queue of all available $NAMB's
;  481          0569  2
;  482          0570  2 $queue_initialize (exch$a_gbl [excg$q_rmsb_use]);       ! Head of queue of all $RMSB's in use
;  483          0571  2 $queue_initialize (exch$a_gbl [excg$q_rmsb_avl]);       ! Head of queue of all available $RMSB's
;  484          0572  2
;  485          0573  2 $queue_initialize (exch$a_gbl [excg$q_rmsb_use]);       ! Head of queue of all $RMSB's in use
;  486          0574  2 $queue_initialize (exch$a_gbl [excg$q_rmsb_avl]);       ! Head of queue of all available $RMSB's
;  487          0575  2
;  488          0576  2 $queue_initialize (exch$a_gbl [excg$q_rt11ctx_use]);   ! Head of queue of all $RT11CTX's in use
;  489          0577  2 $queue_initialize (exch$a_gbl [excg$q_rt11ctx_avl]);   ! Head of queue of all available $RT11CTX's
;  490          0578  2
;  491          0579  2 $queue_initialize (exch$a_gbl [excg$q_volb_use]);       ! Head of queue of all $VOLB's in use
;  492          0580  2 $queue_initialize (exch$a_gbl [excg$q_volb_avl]);       ! Head of queue of all available $VOLB's
;  493          0581  2
;  494          0582  2 ! Init the RMS pointers.  All the RMS blocks are in space between the end of the official SDL defined
;  495          0583  2 ! block and the end of the allocated space.  We will carry a pointer through as we init these RMS pointers.
;  496          0584  2 !
;  497          0585  2 ptr = .exch$a_gbl + excg$k_length;                ! First free byte after SDL structure
;  498          0586  2 exch$a_gbl [excg$a_sysout_fab]  = .ptr;           ! output fab
;  499          0587  2 ptr = .ptr + fab$k_bln;
;  500          0588  2 exch$a_gbl [excg$a_sysout_rab]  = .ptr;           ! output rab
;  501          0589  2 ptr = .ptr + rab$k_bln;
;  502          0590  2 exch$a_gbl [excg$a_sysout_nam]  = .ptr;           ! output nam block
;  503          0591  2 ptr = .ptr + nam$k_bln;
;  504          0592  2 exch$a_gbl [excg$a_sysout_ebuf] = .ptr;          ! output expanded name string
;  505          0593  2 ptr = .ptr + nam$c_maxrss;
;  506          0594  2 exch$a_gbl [excg$a_sysout_rbuf] = .ptr;          ! output result name string
;  507          0595  2
;  508          0596  2 RETURN;
;  509          0597  1 END;


                                    000C 00000     .ENTRY  MAIN_SETUP_CREATE_EXCG, Save R2,R3                      ; 0509
                          53 00000000'  EF   9E 00002     MOVAB   EXCH$A_GBL, R3                                  ;
                          7E      07CA  8F   3C 00009     MOVZWL  #1994, -(SP)                                    ; 0553
                00000000G EF            01   FB 0000E     CALLS   #1, EXCH$UTIL_VM_ALLOCATE_ZEROED                ;
                          63            50   D0 00015     MOVL    R0, EXCH$A_GBL                                  ;
                          51            63   D0 00018     MOVL    EXCH$A_GBL, R1                                  ; 0557
                       08 A1      07CA  8F   B0 0001B     MOVW    #1994, 8(R1)                                    ;
                       0A A1            05   8E 00021     MNEGB   #5, 10(R1)                                      ;
                          50         5C  A1   9E 00025     MOVAB   92(R1), R0                                      ; 0561
                          60            50   D0 00029     MOVL    R0, (R0)                                        ;
                       04 A0            50   D0 0002C     MOVL    R0, 4(R0)                                       ;
                          50         64  A1   9E 00030     MOVAB   100(R1), R0                                     ; 0562
                          60            50   D0 00034     MOVL    R0, (R0)                                        ;
                       04 A0            50   D0 00037     MOVL    R0, 4(R0)                                       ;
                          50         70  A1   9E 0003B     MOVAB   112(R1), R0                                     ; 0564
                          60            50   D0 0003F     MOVL    R0, (R0)                                        ;
                       04 A0            50   D0 00042     MOVL    R0, 4(R0)                                       ;
                          50         78  A1   9E 00046     MOVAB   120(R1), R0                                     ; 0565
                          60            50   D0 0004A     MOVL    R0, (R0)                                        ;
                       04 A0            50   D0 0004D     MOVL    R0, 4(R0)                                       ;
```

```
                    50    0084   C1  9E 00051          MOVAB   132(R1), R0                     ; 0567
                    60           50  D0 00056          MOVL    R0, (R0)
                04  A0           50  D0 00059          MOVL    R0, 4(R0)
                    50    008C   C1  9E 0005D          MOVAB   140(R1), R0                     ; 0568
                    60           50  D0 00062          MOVL    R0, (R0)
                04  A0           50  D0 00065          MOVL    R0, 4(R0)
                    52    0098   C1  9E 00069          MOVAB   152(R1), R2                     ; 0570
                    62           52  D0 0006E          MOVL    R2, (R2)
                04  A2           52  D0 00071          MOVL    R2, 4(R2)
                    50    00A0   C1  9E 00075          MOVAB   160(R1), R0                     ; 0571
                    60           50  D0 0007A          MOVL    R0, (R0)
                04  A0           50  D0 0007D          MOVL    R0, 4(R0)
                    62           52  D0 00081          MOVL    R2, (R2)                        ; 0573
                04  A2           52  D0 00084          MOVL    R2, 4(R2)
                    60           50  D0 00088          MOVL    R0, (R0)                        ; 0574
                04  A0           50  D0 0008B          MOVL    R0, 4(R0)
                    50    00AC   C1  9E 0008F          MOVAB   172(R1), R0                     ; 0576
                    60           50  D0 00094          MOVL    R0, (R0)
                04  A0           50  D0 00097          MOVL    R0, 4(R0)
                    50    00B4   C1  9E 0009B          MOVAB   180(R1), R0                     ; 0577
                    60           50  D0 000A0          MOVL    R0, (R0)
                04  A0           50  D0 000A3          MOVL    R0, 4(R0)
                    50    00C0   C1  9E 000A7          MOVAB   192(R1), R0                     ; 0579
                    60           50  D0 000AC          MOVL    R0, (R0)
                04  A0           50  D0 000AF          MOVL    R0, 4(R0)
                    50    00C8   C1  9E 000B3          MOVAB   200(R1), R0                     ; 0580
                    60           50  D0 000B8          MOVL    R0, (R0)
                04  A0           50  D0 000BB          MOVL    R0, 4(R0)
                    50    01E6   C1  9E 000BF          MOVAB   486(R1), PTR                    ; 0585
            00D0    C1           80  7E 000C4          MOVAQ   (PTR)+, 208(R1)                 ; 0586
                    50      48   A0  9E 000C9          MOVAB   72(R0), PTR                     ; 0587
            00D4    C1           80  7E 000CD          MOVAQ   (PTR)+, 212(R1)                 ; 0588
                    50           3C  C0 000D2          ADDL2   #60, PTR                        ; 0589
            00D8    C1           80  7E 000D5          MOVAQ   (PTR)+, 216(R1)                 ; 0590
                    50      58   A0  9E 000DA          MOVAB   88(R0), PTR                     ; 0591
            00DC    C1           80  7E 000DE          MOVAQ   (PTR)+, 220(R1)                 ; 0592
                    50    00F7   C0  9E 000E3          MOVAB   247(R0), PTR                    ; 0593
            00E0    C1           50  D0 000E8          MOVL    PTR, 224(R1)                    ; 0594
                                 04  000ED          RET                                     ; 0597

; Routine Size:  238 bytes,    Routine Base:  EXCH$MAIN_CODE + 0114
```

```
;   511     0598  1 GLOBAL ROUTINE main_setup_load_time : NOVALUE = %SBTTL 'main_setup_load_time'
;   512     0599  2 BEGIN
;   513     0600  2 !++
;   514     0601  2 !
;   515     0602  2 ! FUNCTIONAL DESCRIPTION:
;   516     0603  2 !
;   517     0604  2 !       This routine performs initializations which are required once only at image load time.
;   518     0605  2 !
;   519     0606  2 ! INPUTS:
;   520     0607  2 !
;   521     0608  2 !       none
;   522     0609  2 !
;   523     0610  2 ! IMPLICIT INPUTS:
;   524     0611  2 !
;   525     0612  2 !       none
;   526     0613  2 !
;   527     0614  2 ! OUTPUTS:
;   528     0615  2 !
;   529     0616  2 !       none
;   530     0617  2 !
;   531     0618  2 ! IMPLICIT OUTPUTS:
;   532     0619  2 !
;   533     0620  2 !       none
;   534     0621  2 !
;   535     0622  2 ! ROUTINE VALUE:
;   536     0623  2 !
;   537     0624  2 !       none
;   538     0625  2 !
;   539     0626  2 ! SIDE EFFECTS:
;   540     0627  2 !
;   541     0628  2 !       none
;   542     0629  2 !--
;   543     0630  2
;   544     0631  2 $dbgtrc_prefix ('main_setup_load_time> ');
;   545     0632  2
;   546     0633  2 LOCAL
;   547     0634  2     dib : $bblock [12],                         ! First longword of dib
;   548     0635  2     dib_desc : VECTOR [2, LONG],                ! A descriptor for the above
;   549     0636  2     jpi_item : VECTOR [10, LONG],               ! Item list for f$getjpi
;   550     0637  2     group,
;   551     0638  2     member,
;   552     0639  2     status
;   553     0640  2     ;
;   554     0641  2
;   555     0642  2 BIND
;   556     0643  2     syscommand = %ASCID 'SYS$COMMAND';
;   557     0644  2
;   558     0645  2 $debug_print_lit ('entry');
;   559     0646  2
;   560     0647  2 ! Allocate and initialize the global data structure
;   561     0648  2 !
;   562     0649  2 main_setup_create_excg ();
;   563     0650  2
;   564     0651  2 ! Now that the global structure is ready, we can bind to some components
;   565     0652  2 !
;   566     0653  3 BEGIN
;   567     0654  3 BIND
```

```
;  568          0655  3      out_fab  = .exch$a_gbl [excg$a_sysout_fab]  : $bblock,
;  569          0656  3      out_rab  = .exch$a_gbl [excg$a_sysout_rab]  : $bblock,
;  570          0657  3      out_nam  = .exch$a_gbl [excg$a_sysout_nam]  : $bblock,
;  571          0658  3      out_ebuf = .exch$a_gbl [excg$a_sysout_ebuf] : $bblock,
;  572          0659  3      out_rbuf = .exch$a_gbl [excg$a_sysout_rbuf] : $bblock
;  573          0660  3      ;
;  574          0661  3
;  575          0662  3  ! Prepare control blocks for terminal I/O
;  576          0663  3  !
;  577        P 0664  3  $fab_init (
;  578        P 0665  3          fab = out_fab,                          ! File access block
;  579        P 0666  3          fac = PUT,                              !  Put only
;  580        P 0667  3          rat = CR,
;  581        P 0668  3          fnm = 'SYS$OUTPUT',
;  582          0669  3          nam = out_nam);                         !  Name block
;  583          0670  3
;  584        P 0671  3  $rab_init (
;  585        P 0672  3          rab = out_rab,                          ! Record access block
;  586        P 0673  3          rac = SEQ,
;  587          0674  3          fab = out_fab);
;  588          0675  3
;  589        P 0676  3  $nam_init (
;  590        P 0677  3          nam = out_nam,                          ! File name block
;  591        P 0678  3          rsa = out_rbuf,                         !  Result name addr
;  592        P 0679  3          rss = nam$c_maxrss,                     !  Result name size
;  593        P 0680  3          esa = out_ebuf,                         !  Expanded name addr
;  594          0681  3          ess = nam$c_maxrss);                    !  Expanded name size
;  595          0682  3
;  596          0683  3  ! Open the default output stream
;  597          0684  3  !
;  598          0685  4  IF NOT (status = $open (fab = out_fab))
;  599          0686  3  THEN
;  600          0687  3      $exit (code = exch$util_file_error (exch$_openout, .status, out_fab, .out_fab [fab$l_stv]));
;  601          0688  3
;  602          0689  4  IF NOT (status = $connect (rab = out_rab))
;  603          0690  3  THEN
;  604          0691  3      $exit (code = exch$util_file_error (exch$_openout, .status, out_fab, .out_rab [rab$l_stv]));
;  605          0692  3
;  606          0693  3  ! If SYS$COMMAND is a terminal device, set up Control/C handlers so that we can interrupt long commands
;  607          0694  3  !
;  608          0695  4  IF NOT (status = $assign (chan=exch$a_gbl [excg$w_tt_channel], devnam=syscommand))
;  609          0696  3  THEN
;  610          0697  3      $exch_signal_stop (exch$_accessfail, 1, syscommand, .status);
;  611          0698  3
;  612          0699  3  ! Get the device information for SYS$COMMAND
;  613          0700  3  !
;  614          0701  3  dib_desc [0] = 12;
;  615          0702  3  dib_desc [1] = dib;
;  616          0703  4  IF NOT (status = $getchn (chan=.exch$a_gbl [excg$w_tt_channel], pribuf=dib_desc))
;  617          0704  3  THEN
;  618          0705  3      $exch_signal_stop (exch$_accessfail, 1, syscommand, .status);
;  619        P 0706  3  $trace_print_fao ('channel !XW, devchar !XL, devclass !XB, devtype !XB, devbufsiz !UL, devdepend !XL',
;  620        P 0707  3                    .exch$a_gbl [excg$w_tt_channel], .dib [dib$l_devchar],
;  621          0708  3                    .dib [dib$b_devclass], .dib [dib$b_devtype], .dib [dib$w_devbufsiz], .dib [dib$l_devdepend])
;  622          0709  3
;  623          0710  3  ! If SYS$COMMAND is a terminal, enable the control/c ast
;  624          0711  3  !
```

```
;   625          0712  3 IF .dib [dib$b_devclass] EQL dc$_term
;   626          0713  3 THEN
;   627          0714  4     BEGIN
;   628          0715  4     LOCAL
;   629          0716  4         iosb : VECTOR [4, WORD];
;   630          0717  4
;   631          0718  4     ! Set the control/c ast, renabling it to this routine
;   632          0719  4     !
;   633          0720  4     $trace_print_lit ('SYS$COMMAND is a terminal, enabling control/c');
;   634        P 0721  5     IF (status = $qiow (efn=0, chan=.exch$a_gbl [excg$w_tt_channel],
;   635          0722  5                         func=(io$_setmode OR io$m_outband), iosb=iosb, p1=main_control_c_ast, p2=UPLIT (0,8)))
;   636          0723  4     THEN
;   637          0724  4         status = .iosb [0];
;   638          0725  4
;   639          0726  4     ! If either the qio or the io operation failed, scream and shout
;   640          0727  4     !
;   641          0728  4     IF NOT .status
;   642          0729  4     THEN
;   643          0730  4         $exch_signal_stop (.status);
;   644          0731  4     END
;   645          0732  3 ELSE
;   646          0733  4     BEGIN
;   647          0734  4     $trace_print_lit ('SYS$COMMAND is not a terminal, no control/c');
;   648          0735  4     $dassgn (chan = .exch$a_gbl [excg$w_tt_channel]);   ! Deassign the channel, we have no further use for i
;   649          0736  4     exch$a_gbl [excg$w_tt_channel] = 0;                 ! Mark channel as not in use
;   650          0737  3     END;
;   651          0738  3
;   652          0739  3 ! Get the user's UIC group and member numbers, in case we create any files.
;   653          0740  3 !
;   654          0741  3 %IF switch_variant GEQ 3 %THEN group = member = 0; %FI! While debugging, suppress the bogus "uninit referenc
;   655          0742  3 jpi_item [0] = (jpi$_grp^16 OR 4);                   ! Group number
;   656          0743  3 jpi_item [1] = group;                               ! Buffer for value
;   657          0744  3 jpi_item [2] = 0;                                   ! Returned length not important
;   658          0745  3 jpi_item [3] = (jpi$_mem^16 OR 4);                   ! Member number
;   659          0746  3 jpi_item [4] = member;
;   660          0747  3 jpi_item [5] = 0;
;   661          0748  3 jpi_item [6] = (jpi$_username^16 OR 12);
;   662          0749  3 jpi_item [7] = exch$a_gbl [excg$t_username];
;   663          0750  3 jpi_item [8] = 0;
;   664          0751  3 jpi_item [9] = 0;                                   ! End of list
;   665          0752  3
;   666          0753  4 IF NOT (status = $getjpiw (efn=0, itmlst=jpi_item))
;   667          0754  3 THEN
;   668          0755  3     $exch_signal_stop (.status);
;   669          0756  3 exch$a_gbl [excg$w_uic_group]  = .group;
;   670          0757  3 exch$a_gbl [excg$w_uic_member] = .member;
;   671          0758  3
;   672          0759  3 ! Get the value of command line qualifiers which last for the life of the image
;   673          0760  3 !
;   674          0761  3 exch$a_gbl [excg$v_q_message] = cli$present (%ASCID 'MESSAGE');
;   675          0762  3
;   676          0763  3 ! If global caching is requested, set up the exit handler
;   677          0764  3 !
;   678          0765  4 IF (exch$a_gbl [excg$v_q_cache] = cli$present (%ASCID 'CACHE'))
;   679          0766  3 THEN
;   680          0767  4     BEGIN
;   681          0768  4     $trace_print_lit ('caching requested');
```

```
;  682          0769  4     exch$a_gbl [excg$a_cachexh_routine]   = main_exit_handler;                              ! Address of routine
;  683          0770  4     exch$a_gbl [excg$l_cachexh_arg_count] = 1;                                               ! Number of args (st
;  684          0771  4     exch$a_gbl [excg$a_cachexh_status]    = exch$a_gbl [excg$l_cachexh_condvalu];            ! Location for statu
;  685          0772  4
;  686          0773  5     IF NOT (status = $dclexh (desblk=exch$a_gbl [excg$r_cachexit_block]))
;  687          0774  4     THEN
;  688          0775  4         $exch_signal_stop (.status);
;  689          0776  3     END;
;  690          0777  3
;  691          0778  2 END;                                                 ! Extra end needed for the BIND
;  692          0779  2 RETURN;
;  693          0780  1 END;
; INFO#250                 L1:0756
; Referenced LOCAL symbol GROUP is probably not initialized
; INFO#250                 L1:0757
; Referenced LOCAL symbol MEMBER is probably not initialized


                                                        .PSECT  EXCH$MAIN_PLIT,NOWRT,2

          00  44  4E  41  4D  4D  4F  43  24  53  59  53  0002C P.AAF:   .ASCII  \SYS$COMMAND\<0>                    ;
                                          010E000B  00038 P.AAE:   .LONG   17694731                                ;
                                          00000000' 0003C          .ADDRESS P.AAF                                  ;
                  54  55  50  54  55  4F  24  53  59  53  00040 P.AAG:   .ASCII  \SYS$OUTPUT\                       ;
                                                  0004A            .BLKB   2                                       ;
                              00000008  00000000  0004C P.AAH:   .LONG   0, 8                                     ;
                          00  45  47  41  53  53  45  4D  00054 P.AAJ:   .ASCII  \MESSAGE\<0>                      ;
                                          010E0007  0005C P.AAI:   .LONG   17694727                                ;
                                          00000000' 00060          .ADDRESS P.AAJ                                  ;
                          00  00  00  45  48  43  41  43  00064 P.AAL:   .ASCII  \CACHE\<0><0><0>                   ;
                                          010E0005  0006C P.AAK:   .LONG   17694725                                ;
                                          00000000' 00070          .ADDRESS P.AAL                                  ;

                                                SYSCOMMAND=         P.AAE
                                                        .EXTRN  SYS$OPEN, SYS$CONNECT
                                                        .EXTRN  SYS$ASSIGN, EXCH$_ACCESSFAIL
                                                        .EXTRN  SYS$GETCHN, SYS$QIOW
                                                        .EXTRN  SYS$DASSGN, SYS$GETJPIW
                                                        .EXTRN  CLI$PRESENT, SYS$DCLEXH

                                                        .PSECT  EXCH$MAIN_CODE,NOWRT,2

                                          0FFC 00000            .ENTRY  MAIN_SETUP_LOAD_TIME, Save R2,R3,R4,R5,R6,- ; 0598
                                                                        R7,R8,R9,R10,R11                              ;
                      5B      0000'  CF   9E 00002              MOVAB   SYSCOMMAND, R11                             ;
                      5A 00000000'   EF   9E 00007              MOVAB   EXCH$A_GBL, R10                             ;
                      5E         B4  AE   9E 0000E              MOVAB   -76(SP), SP                                 ;
                FEFB  CF             00   FB 00012              CALLS   #0, MAIN_SETUP_CREATE_EXCG                  ; 0649
                      58             6A   D0 00017              MOVL    EXCH$A_GBL, R8                              ; 0655
                      57      00D0   C8   D0 0001A              MOVL    208(R8), R7                                 ;
                      59      00D4   C8   D0 0001F              MOVL    212(R8), R9                                 ; 0656
                      56      00D8   C8   D0 00024              MOVL    216(R8), R6                                 ; 0657
          0050  8F          00         6E   00   2C 00029       MOVC5   #0, (SP), #0, #80, (R7)                     ; 0669
                                               67   00030                                                           ;
                      67      5003   8F   B0 00031              MOVW    #20483, (R7)                                ;
                    16        A7         01   90 00036          MOVB    #1, 22(R7)                                  ;
```

```
                                1E   A7      0202    8F   B0 0003A          MOVW    #514, 30(R7)                                    ;
                                28   A7              56   D0 00040          MOVL    R6, 40(R7)                                      ;
                                2C   A7        08    AB   9E 00044          MOVAB   P.AAG, 44(R7)                                   ;
                                34   A7              0A   90 00049          MOVB    #10, 52(R7)                                     ;
    0044   8F          00            6E              00   2C 0004D          MOVC5   #0, (SP), #0, #68, (R9)                         ; 0674
                                                    69       00054
                                     69      4401    8F   B0 00055          MOVW    #17409, (R9)                                    ;
                                       1E            A9   94 0005A          CLRB    30(R9)                                          ;
                                3C   A9              57   D0 0005D          MOVL    R7, 60(R9)                                      ;
    0060   8F          00            6E              00   2C 00061          MOVC5   #0, (SP), #0, #96, (R6)                         ; 0681
                                                    66       00068
                                     66      6002    8F   B0 00069          MOVW    #24578, (R6)                                    ;
                                02   A6              01   8E 0006E          MNEGB   #1, 2(R6)                                       ;
                                04   A6      00E0    C8   D0 00072          MOVL    224(R8), 4(R6)                                  ;
                                0A   A6              01   8E 00078          MNEGB   #1, 10(R6)                                      ;
                                0C   A6      00DC    C8   D0 0007C          MOVL    220(R8), 12(R6)                                 ;
                                                     57   DD 00082          PUSHL   R7                                              ; 0685
                          00000000G  00              01   FB 00084          CALLS   #1, SYS$OPEN                                    ;
                                     53              50   D0 0008B          MOVL    R0, STATUS                                      ;
                                     1D              53   E8 0008E          BLBS    STATUS, 1$                                      ;
                                                     A7   DD 00091          PUSHL   12(R7)                                          ; 0687
                                       0088          8F   BB 00094          PUSHR   #^M<R3,R7>                                      ;
                                00F810A0             8F   DD 00098          PUSHL   #16257184                                       ;
                          00000000G  EF              04   FB 0009E          CALLS   #4, EXCH$UTIL_FILE_ERROR                        ;
                                                     50   DD 000A5          PUSHL   R0                                              ;
                          00000000G  00              01   FB 000A7          CALLS   #1, SYS$EXIT                                    ;
                                                     59   DD 000AE 1$:      PUSHL   R9                                              ; 0689
                          00000000G  00              01   FB 000B0          CALLS   #1, SYS$CONNECT                                 ;
                                     53              50   D0 000B7          MOVL    R0, STATUS                                      ;
                                     1D              53   E8 000BA          BLBS    STATUS, 2$                                      ;
                                                     A9   DD 000BD          PUSHL   12(R9)                                          ; 0691
                                       0088          8F   BB 000C0          PUSHR   #^M<R3,R7>                                      ;
                                00F810A0             8F   DD 000C4          PUSHL   #16257184                                       ;
                          00000000G  EF              04   FB 000CA          CALLS   #4, EXCH$UTIL_FILE_ERROR                        ;
                                                     50   DD 000D1          PUSHL   R0                                              ;
                          00000000G  00              01   FB 000D3          CALLS   #1, SYS$EXIT                                    ;
                                                     7E   7C 000DA 2$:      CLRQ    -(SP)                                           ; 0695
    7E                           6A                 02   C1 000DC          ADDL3   #2, EXCH$A_GBL, -(SP)                           ;
                                                     5B   DD 000E0          PUSHL   R11                                             ;
                          00000000G  00              04   FB 000E2          CALLS   #4, SYS$ASSIGN                                  ;
                                     53              50   D0 000E9          MOVL    R0, STATUS                                      ;
                                     24              53   E9 000EC          BLBC    STATUS, 3$                                      ;
                                38   AE              0C   D0 000EF          MOVL    #12, DIB_DESC                                   ; 0701
                                3C   AE        40    AE   9E 000F3          MOVAB   DIB, DIB_DESC+4                                 ; 0702
                                                     7E   7C 000F8          CLRQ    -(SP)                                           ; 0703
                                       40            AE   9F 000FA          PUSHAB  DIB_DESC                                        ;
                                                     7E   D4 000FD          CLRL    -(SP)                                           ;
                                     50              6A   D0 000FF          MOVL    EXCH$A_GBL, R0                                  ;
                                     7E        02    A0   3C 00102          MOVZWL  2(R0), -(SP)                                    ;
                          00000000G  00              05   FB 00106          CALLS   #5, SYS$GETCHN                                  ;
                                     53              50   D0 0010D          MOVL    R0, STATUS                                      ;
                                     14              53   E8 00110          BLBS    STATUS, 4$                                      ;
                                                     53   DD 00113 3$:      PUSHL   STATUS                                          ; 0705
                                                     5B   DD 00115          PUSHL   R11                                             ;
                                                     01   DD 00117          PUSHL   #1                                              ;
                                       00000000G     8F   DD 00119          PUSHL   #EXCH$_ACCESSFAIL                               ;
                          00000000G  00              04   FB 0011F          CALLS   #4, LIB$STOP                                    ;
```

```
                                        04 00126                RET
                                50      D0 00127 4$:            MOVL    EXCH$A_GBL, R0                          ; 0722
                            42  8F  44 AE 91 0012A             CMPB    DIB+4, #66                              ; 0712
                                        32 12 0012F             BNEQ    6$
                                        7E 7C 00131             CLRQ    -(SP)                                   ; 0722
                                        7E 7C 00133             CLRQ    -(SP)
                                    14  AB 9F 00135             PUSHAB  P.AAH
                                  FCC2  CF 9F 00138             PUSHAB  MAIN_CONTROL_C_AST
                                        7E 7C 0013C             CLRQ    -(SP)
                                    28  AE 9F 0013E             PUSHAB  IOSB
                                7E  0423  8F 3C 00141           MOVZWL  #1059, -(SP)
                                7E    02  A0 3C 00146           MOVZWL  2(R0), -(SP)
                                        7E D4 0014A             CLRL    -(SP)
                    00000000G   00      0C FB 0014C             CALLS   #12, SYS$QIOW
                                53      50 D0 00153             MOVL    R0, STATUS
                                07      53 E9 00156             BLBC    STATUS, 5$
                                53  08  AE 3C 00159             MOVZWL  IOSB, STATUS                            ; 0724
                                14      53 E8 0015D             BLBS    STATUS, 7$                              ; 0728
                                    00A8 31 00160 5$:           BRW     8$                                      ; 0730
                                7E    02  A0 3C 00163 6$:       MOVZWL  2(R0), -(SP)                            ; 0735
                    00000000G   00      01 FB 00167             CALLS   #1, SYS$DASSGN
                                50      6A D0 0016E             MOVL    EXCH$A_GBL, R0                          ; 0736
                                    02  A0 B4 00171             CLRW    2(R0)
                            10  AE 03080004  8F D0 00174 7$:    MOVL    #50855940, JPI_ITEM                     ; 0742
                            14  AE          6E 9E 0017C         MOVAB   GROUP, JPI_ITEM+4                       ; 0743
                                    18  AE D4 00180             CLRL    JPI_ITEM+8                              ; 0744
                            1C  AE 03070004  8F D0 00183        MOVL    #50790404, JPI_ITEM+12                  ; 0745
                            20  AE      04  AE 9E 0018B         MOVAB   MEMBER, JPI_ITEM+16                     ; 0746
                                    24  AE D4 00190             CLRL    JPI_ITEM+20                             ; 0747
                            28  AE 0202000C  8F D0 00193        MOVL    #33685516, JPI_ITEM+24                  ; 0748
                        2C  AE          6A  20 C1 0019B         ADDL3   #32, EXCH$A_GBL, JPI_ITEM+28            ; 0749
                                    30  AE 7C 001A0             CLRQ    JPI_ITEM+32                             ; 0750
                                        7E 7C 001A3             CLRQ    -(SP)                                   ; 0753
                                        7E D4 001A5             CLRL    -(SP)
                                    1C  AE 9F 001A7             PUSHAB  JPI_ITEM
                                        7E 7C 001AA             CLRQ    -(SP)
                                        7E D4 001AC             CLRL    -(SP)
                    00000000G   00      07 FB 001AE             CALLS   #7, SYS$GETJPIW
                                53      50 D0 001B5             MOVL    R0, STATUS
                                50      53 E9 001B8             BLBC    STATUS, 8$
                                52      6A D0 001BB             MOVL    EXCH$A_GBL, R2                          ; 0756
                            1E  A2      6E B0 001BE             MOVW    GROUP, 30(R2)
                            1C  A2  04  AE B0 001C2             MOVW    MEMBER, 28(R2)                          ; 0757
                                    24  AB 9F 001C7             PUSHAB  P.AAI                                   ; 0761
                    00000000G   00      01 FB 001CA             CALLS   #1, CLI$PRESENT
                    62              01  02 F0 001D1            INSV    R0, #2, #1, (R2)
                                    34  AB 9F 001D6             PUSHAB  P.AAK                                   ; 0765
                    00000000G   00      01 FB 001D9             CALLS   #1, CLI$PRESENT
                00  BA              01  01 F0 001E0            INSV    R0, #1, #1, @EXCH$A_GBL
                                        2B 50 E9 001E6          BLBC    R0, 9$
                                50      6A D0 001E9             MOVL    EXCH$A_GBL, R0                          ; 0769
                            48  A0  FC30  CF 9E 001EC            MOVAB   MAIN_EXIT_HANDLER, 72(R0)
                            4C  A0      01 D0 001F2             MOVL    #1, 76(R0)                              ; 0770
                            50  A0  54  A0 9E 001F6             MOVAB   84(R0), 80(R0)                          ; 0771
                                44  A0 9F 001FB                 PUSHAB  68(R0)                                  ; 0773
                    00000000G   00      01 FB 001FE             CALLS   #1, SYS$DCLEXH
                                53      50 D0 00205             MOVL    R0, STATUS
```

```
                          09        53  E8 00208          BLBS    STATUS, 9$                              ;
                                    53  DD 0020B 8$:      PUSHL   STATUS                                  ; 0775
                00000000G 00        01  FB 0020D          CALLS   #1, LIB$STOP                            ;
                                        04 00214 9$:      RET                                             ; 0780

; Routine Size:  533 bytes,    Routine Base:  EXCH$MAIN_CODE + 0202
```

```
;  695     0781  1 GLOBAL ROUTINE main_start =                %SBTTL 'main_start'
;  696     0782  2 BEGIN
;  697     0783  2 !++
;  698     0784  2 !
;  699     0785  2 ! FUNCTIONAL DESCRIPTION:
;  700     0786  2 !
;  701     0787  2 !       Main procedure of EXCHANGE.  Contains main command input loop.
;  702     0788  2 !
;  703     0789  2 ! INPUTS:
;  704     0790  2 !
;  705     0791  2 !       none
;  706     0792  2 !
;  707     0793  2 ! IMPLICIT INPUTS:
;  708     0794  2 !
;  709     0795  2 !       Invoking command line if present, otherwise none.
;  710     0796  2 !
;  711     0797  2 ! OUTPUTS:
;  712     0798  2 !
;  713     0799  2 !       None
;  714     0800  2 !
;  715     0801  2 ! IMPLICIT OUTPUTS:
;  716     0802  2 !
;  717     0803  2 !       none
;  718     0804  2 !
;  719     0805  2 ! ROUTINE VALUE:
;  720     0806  2 !
;  721     0807  2 !       true, or error code if abnormal termination
;  722     0808  2 !
;  723     0809  2 ! SIDE EFFECTS:
;  724     0810  2 !
;  725     0811  2 !       Many.
;  726     0812  2 !--
;  727     0813  2
;  728     0814  2 $dbgtrc_prefix ('main_start> ');
;  729     0815  2
;  730     0816  2 LOCAL
;  731     0817  2     dynamic_desc        : $dyn_str_desc,        ! A dynamic string descriptor for "foreign" commands
;  732     0818  2     status
;  733     0819  2     ;
;  734     0820  2
;  735     0821  2 ! The CLI will print an annoying "%CLI-W-NOCOMD, no command on line" message if a blank line is entered.  We
;  736     0822  2 ! declare a condition handler which will stop such nonsense.
;  737     0823  2 !
;  738     0824  2 ENABLE
;  739     0825  2     main_handle_cli_nocomd;
;  740     0826  2
;  741     0827  2 ! Check that some of our constants have valid values.  Note that the $logic_check macro will perform checks
;  742     0828  2 ! compile-time rather than run-time if possible.
;  743     0829  2 !
;  744   L 0830  2 $logic_check (0, ((ctx$k_buffer_length GEQU 1536) AND (ctx$k_buffer_length LSSU 65536)), 114);
; %PRINT:       assumption 114 verified during compilation
;  745   L 0831  2 $logic_check (0, (ctx$k_buffer_length GEQU filb$s_record_buffer+1024), 143);
; %PRINT:       assumption 143 verified during compilation
;  746   L 0832  2 $logic_check (0, (rt11ctx$s_entry EQL rt11ent$k_length), 167);
; %PRINT:       assumption 167 verified during compilation
;  747   L 0833  2 $logic_check (0, (dos11ctx$s_entry_fields EQL dos11ctx$s_entry), 251);
; %PRINT:       assumption 251 verified during compilation
```

```
;  748          L 0834  2 $logic_check (0, (rt11hom$k_length EQL 512), 141);
; %PRINT:          assumption 141 verified during compilation
;  749          L 0835  2 $logic_check (0, (rt11$s_home_block EQL 512), 292);
; %PRINT:          assumption 292 verified during compilation
;  750             0836  2
;  751             0837  2 ! Several routines assume that the fields in the front of the CTX$, RT11CTX$ and DOS11CTX$ structures
;  752             0838  2 ! coincide.  Test these assumptions (again, compile-time checks).
;  753             0839  2 !
;  754          L 0840  2 $logic_check (0, ($$offset_check (a_alloc)), 293);
; %PRINT:          assumption 293 verified during compilation
;  755          L 0841  2 $logic_check (0, ($$offset_check (a_assoc_filb)), 294);
; %PRINT:          assumption 294 verified during compilation
;  756          L 0842  2 $logic_check (0, ($$offset_check (a_assoc_volb)), 295);
; %PRINT:          assumption 295 verified during compilation
;  757          L 0843  2 $logic_check (0, ($$offset_check (a_buffer)), 296);
; %PRINT:          assumption 296 verified during compilation
;  758          L 0844  2 $logic_check (0, ($$offset_check (l_cur_block)), 297);
; %PRINT:          assumption 297 verified during compilation
;  759          L 0845  2 $logic_check (0, ($$offset_check (l_eof_block)), 298);
; %PRINT:          assumption 298 verified during compilation
;  760          L 0846  2 $logic_check (0, ($$offset_check (l_cur_byte)), 299);
; %PRINT:          assumption 299 verified during compilation
;  761          L 0847  2 $logic_check (0, ($$offset_check (l_flags)), 300);
; %PRINT:          assumption 300 verified during compilation
;  762          L 0848  2 $logic_check (0, ($$offset_check (l_buf_base_block)), 301);
; %PRINT:          assumption 301 verified during compilation
;  763          L 0849  2 $logic_check (0, ($$offset_check (l_buf_high_block)), 302);
; %PRINT:          assumption 302 verified during compilation
;  764          L 0850  2 $logic_check (0, ($$offset_check (l_high_block_written)), 303);
; %PRINT:          assumption 303 verified during compilation
;  765          L 0851  2 $logic_check (0, ($$bit_check (v_stream_active)), 304);
; %PRINT:          assumption 304 verified during compilation
;  766          L 0852  2 $logic_check (0, ($$bit_check (v_output_file)), 305);
; %PRINT:          assumption 305 verified during compilation
;  767          L 0853  2 $logic_check (0, ($$bit_check (v_flush)), 306);
; %PRINT:          assumption 306 verified during compilation
;  768             0854  2
;  769             0855  2 ! Perform initializations necessary only once at load time.  SYS$OUTPUT is attached to SYSOUT_xAB.
;  770             0856  2 ! Qualifiers on the EXCHANGE verb itself are parsed and recorded in global variables.
;  771             0857  2 !
;  772             0858  2 main_setup_load_time ();                          ! If failed, exit to VMS.
```

```
;  774     0859  2 !+
;  775     0860  2 ! Files have been initialized.  If executed as a foreign command, perform requested function and exit.
;  776     0861  2 !-
;  777     0862  2
;  778     0863  2 IF cli$get_value (%ASCID 'COMMAND', dynamic_desc)
;  779     0864  2 THEN
;  780     0865  3     BEGIN
;  781     0866  3
;  782     0867  3     ! Parse the single command, execute if successful
;  783     0868  3     !
;  784     0869  3     exch$a_gbl [excg$v_foreign_command] = true;
;  785     0870  4     IF (status = cli$dcl_parse (dynamic_desc, exch$cld_table))
;  786     0871  4     THEN
;  787     0872  4         status = cli$dispatch();
;  788     0873  3
;  789     0874  3     $check_call (4, lib$signal, exch$_trace, 1, .status, .status);
;  790     0875  3
;  791     0876  3     ! Terminate execution and return to DCL
;  792     0877  3     !
;  793     0878  3     IF NOT .status                              ! If failed, inhibit additional signalling
;  794     0879  3     THEN
;  795     0880  3         $inhibit_msg (status);
;  796     0881  3
;  797     0882  3     RETURN .status;
;  798     0883  2     END;
```

```
;  800          0884  2 !+
;  801          0885  2 ! Top of the interactive command loop.  The normal exit condition is a call to exch$main_exit, which occurs
;  802          0886  2 ! end-of-file is reached on the SYS$INPUT stream or the verb EXIT is received.
;  803          0887  2 !-
;  804          0888  2 DO
;  805          0889  3     BEGIN
;  806          0890  3
;  807          0891  3
;  808          0892  3     ! Call the library routine to parse the command, pass the address of the external command table
;  809          0893  3     !
;  810          0894  4     IF (status = cli$dcl_parse (0, exch$cld_table, lib$get_input, lib$get_input, %ASCID 'EXCHANGE> '))
;  811          0895  3     THEN
;  812          0896  4         BEGIN
;  813          0897  4
;  814          0898  4         exch$a_gbl [excg$v_control_c] = false;  ! Clear the bit saying we got an ast
;  815          0899  4
;  816          0900  4         ! Call the library routine to execute (call) the routine associated with the DCL verb
;  817          0901  4         !
;  818          0902  4         status = cli$dispatch();
;  819          0903  4
;  820          0904  4         ! Keep track of status during development
;  821          0905  4         !
;  822          0906  4         $check_call (4, lib$signal, exch$_trace, 1, .status, .status);
;  823          0907  4
;  824          0908  3         END;
;  825          0909  3
;  826          0910  3     END
;  827          0911  2 UNTIL .status EQL rms$_eof;
;  828          0912  2
;  829          0913  2 RETURN true;
;  830          0914  1 END;


                                                        .PSECT  EXCH$MAIN_PLIT,NOWRT,2

                  00  44  4E  41  4D  4D  4F  43  00074 P.AAN:  .ASCII  \COMMAND\<0>                                    ;
                                    010E0007  0007C P.AAM:  .LONG   17694727                                          ;
                                    00000000' 00080         .ADDRESS P.AAN                                            ;
00  00  20  3E  45  47  4E  41  48  43  58  45  00084 P.AAP:  .ASCII  \EXCHANGE> \<0><0>                                ;
                                    010E000A  00090 P.AAO:  .LONG   17694730                                          ;
                                    00000000' 00094         .ADDRESS P.AAP                                            ;

                                                        .EXTRN  CLI$DCL_PARSE, EXCH$CLD_TABLE
                                                        .EXTRN  CLI$DISPATCH, EXCH$_TRACE

                                                        .PSECT  EXCH$MAIN_CODE,NOWRT,2

                                        003C 00000      .ENTRY  MAIN_START, Save R2,R3,R4,R5                              ; 0781
                55 00000000G   00    9E 00002           MOVAB   LIB$GET_INPUT, R5                                         ;
                54 00000000G   00    9E 00009           MOVAB   CLI$DISPATCH, R4                                          ;
                53 00000000G   00    9E 00010           MOVAB   CLI$DCL_PARSE, R3                                         ;
                52 00000000G   EF    9E 00017           MOVAB   EXCH$CLD_TABLE, R2                                        ;
                5E             04    C2 0001E           SUBL2   #4, SP                                                    ;
                   020E0000    8F    DD 00021           PUSHL   #34471936                                                 ; 0817
                         04    AE    D4 00027           CLRL    DYNAMIC_DESC+4                                            ;
                6D     005B    CF    DE 0002A           MOVAL   5$, (FP)                                                  ;
```

```
                         FDB7   CF              00 FB 0002F          CALLS   #0, MAIN_SETUP_LOAD_TIME                ; 0858
                                                5E DD 00034          PUSHL   SP                                      ; 0863
                                          0000' CF 9F 00036          PUSHAB  P.AAM                                   ;
                    00000000G   00              02 FB 0003A          CALLS   #2, CLI$GET_VALUE                       ;
                                1E              50 E9 00041          BLBC    R0, 2$                                  ;
                    00000000'   FF              08 88 00044          BISB2   #8, @EXCH$A_GBL                         ; 0869
                                                52 DD 0004B          PUSHL   R2                                      ; 0870
                                          04    AE 9F 0004D          PUSHAB  DYNAMIC_DESC                            ;
                                63              02 FB 00050          CALLS   #2, CLI$DCL_PARSE                       ;
                                06              50 E9 00053          BLBC    STATUS, 1$                              ;
                                64              00 FB 00056          CALLS   #0, CLI$DISPATCH                        ; 0872
                                2C              50 E8 00059          BLBS    STATUS, 4$                              ; 0878
          50        01                  1C              01 F0 0005C 1$:  INSV    #1, #28, #1, STATUS2                    ; 0880
                                                        04 00061     RET                                             ; 0882
                                          0000' CF 9F 00062 2$:      PUSHAB  P.AAO                                   ; 0894
                                                55 DD 00066          PUSHL   R5                                      ;
                                                24 BB 00068          PUSHR   #^M<R2,R5>                              ;
                                                7E D4 0006A          CLRL    -(SP)                                   ;
                                63              05 FB 0006C          CALLS   #5, CLI$DCL_PARSE                       ;
                                0A              50 E9 0006F          BLBC    STATUS, 3$                              ;
                    00000000'   FF              01 8A 00072          BICB2   #1, @EXCH$A_GBL                         ; 0898
                                64              00 FB 00079          CALLS   #0, CLI$DISPATCH                        ; 0902
                    0001827A    8F              50 D1 0007C 3$:      CMPL    STATUS, #98938                          ; 0911
                                0D              12 00083             BNEQ    2$                                      ;
                                50              01 D0 00085          MOVL    #1, R0                                  ; 0913
                                                    04 00088 4$:     RET                                             ; 0914
                                                  0000 00089 5$:     .WORD   Save nothing                            ; 0817
                                                7E D4 0008B          CLRL    -(SP)                                   ;
                                                5E DD 0008D          PUSHL   SP                                      ;
                                7E        04    AC 7D 0008F          MOVQ    4(AP), -(SP)                            ;
                         FBF6   CF              03 FB 00093          CALLS   #3, MAIN_HANDLE_CLI_NOCOMD              ;
                                                    04 00098         RET                                             ;

; Routine Size:  153 bytes,    Routine Base:  EXCH$MAIN_CODE + 0417
```

```
;  832          0915  1 END                          !End of module
;  833          0916  0 ELUDOM



                                                      .EXTRN  LIB$SIGNAL

;                              PSECT SUMMARY
;
;       Name                    Bytes                   Attributes
;
;  EXCH$MAIN_PLIT                152  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  EXCH$RW_GLOBAL                  4  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  EXCH$MAIN_CODE               1200  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                           Library Statistics
;                                    -------- Symbols --------      Pages       Processing
;       File                         Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1      18619       126         0        1000        00:01.9
;  _$255$DUA28:[EXCHNG.OBJ]EXCLIB.L32;1  1151       130        11          79        00:01.3



; Information:  2
; Warnings:     0
; Errors:       0


;                                    COMMAND QUALIFIERS

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

; Size:          1200 code + 156 data bytes
; Run Time:        00:32.1
; Elapsed Time:    01:36.7
; Lines/CPU Min:    1713
; Lexemes/CPU-Min: 35144
; Memory Used:  248 pages
; Compilation Complete
```

EXCRT11.
LIS

EXCMAIN
LIS

EXCMSG
LIS

EXCPDP.
LIS

EXCMOUN
LIS