```
   SSSSSSSSSSS     000000000     RRRRRRRRRRR     TTTTTTTTTTTTTTT     333333333       222222222
   SSSSSSSSSSS     000000000     RRRRRRRRRRRR    TTTTTTTTTTTTTTT     333333333       222222222
   SSSSSSSSSSS     000000000     RRRRRRRRRRRR    TTTTTTTTTTTTTTT     333333333       222222222
SSS              000          000  RRR         RRR        TTT        333          333  222          222
SSS              000          000  RRR         RRR        TTT        333          333  222          222
SSS              000          000  RRR         RRR        TTT        333          333  222          222
SSS              000          000  RRR         RRR        TTT                     333                222
SSS              000          000  RRR         RRR        TTT                     333                222
SSS              000          000  RRR         RRR        TTT                     333                222
   SSSSSSSSS     000          000  RRRRRRRRRRR            TTT                333                   222
   SSSSSSSSS     000          000  RRRRRRRRRRRR           TTT                333                   222
   SSSSSSSSS     000          000  RRRRRRRRRRRR           TTT                333                   222
            SSS  000          000  RRR   RRR              TTT                   333             222
            SSS  000          000  RRR   RRR              TTT                   333             222
            SSS  000          000  RRR   RRR              TTT                   333             222
            SSS  000          000  RRR      RRR           TTT        333        333          222
            SSS  000          000  RRR      RRR           TTT        333        333          222
            SSS  000          000  RRR      RRR           TTT        333        333          222
            SSS  000          000  RRR      RRR           TTT        333        333          222
SSSSSSSSSSS          000000000     RRR         RRR        TTT           333333333       222222222222222
SSSSSSSSSSS          000000000     RRR         RRR        TTT           333333333       222222222222222
SSSSSSSSSSS          000000000     RRR         RRR        TTT           333333333       222222222222222
```

```
UU      UU  TTTTTTTTTT  LL            CCCCCCCC  HH      HH  KK      KK    CCCCCCCC  LL            IIIIII
UU      UU  TTTTTTTTTT  LL            CCCCCCCC  HH      HH  KK      KK    CCCCCCCC  LL            IIIIII
UU      UU      TT      LL          CC          HH      HH  KK      KK  CC          LL              II
UU      UU      TT      LL          CC          HH      HH  KK      KK  CC          LL              II
UU      UU      TT      LL          CC          HH      HH  KK    KK    CC          LL              II
UU      UU      TT      LL          CC          HH      HH  KK    KK    CC          LL              II
UU      UU      TT      LL          CC          HHHHHHHHHH  KKKKKK      CC          LL              II
UU      UU      TT      LL          CC          HHHHHHHHHH  KKKKKK      CC          LL              II
UU      UU      TT      LL          CC          HH      HH  KK    KK    CC          LL              II
UU      UU      TT      LL          CC          HH      HH  KK    KK    CC          LL              II
UU      UU      TT      LL          CC          HH      HH  KK      KK  CC          LL              II          ....
UU      UU      TT      LL            CCCCCCCC  HH      HH  KK      KK    CCCCCCCC  LL            IIIIII        ....
UUUUUUUUUU      TT      LLLLLLLLLL    CCCCCCCC  HH      HH  KK      KK    CCCCCCCC  LLLLLLLLLL    IIIIII        ....
UUUUUUUUUU      TT      LLLLLLLLLL    CCCCCCCC  HH      HH  KK      KK    CCCCCCCC  LLLLLLLLLL    IIIIII        ....

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 UTIL$CHK_CLI (
;     2          0002  0                 IDENT = 'V04-000'       ! File: UTLCHKCLI.B32 Edit: PDG3001
;     3          0003  0                 ) =
;     4          0004  1 BEGIN
;     5          0005  1 !
;     6          0006  1 !**************************************************************************
;     7          0007  1 !*                                                                        *
;     8          0008  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                              *
;     9          0009  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                *
;    10          0010  1 !*  ALL RIGHTS RESERVED.                                                  *
;    11          0011  1 !*                                                                        *
;    12          0012  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
;    13          0013  1 !*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE        *
;    14          0014  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER   *
;    15          0015  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
;    16          0016  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY     *
;    17          0017  1 !*  TRANSFERRED.                                                           *
;    18          0018  1 !*                                                                        *
;    19          0019  1 !*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE   *
;    20          0020  1 !*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT       *
;    21          0021  1 !*  CORPORATION.                                                           *
;    22          0022  1 !*                                                                        *
;    23          0023  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS   *
;    24          0024  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                *
;    25          0025  1 !*                                                                        *
;    26          0026  1 !*                                                                        *
;    27          0027  1 !**************************************************************************
;    28          0028  1 !
;    29          0029  1
;    30          0030  1 !++
;    31          0031  1 !
;    32          0032  1 ! FACILITY:     VAX-11 SORT/MERGE
;    33          0033  1 !               VAX-11 CHECKPOINT/RESTART
;    34          0034  1 !               VAX-11 CLI
;    35          0035  1 !
;    36          0036  1 ! ABSTRACT:
;    37          0037  1 !
;    38          0038  1 !       This module contains routines that can be used by a checkpointed
;    39          0039  1 !       subprocess to obtain CLI information from a parent process.
;    40          0040  1 !
;    41          0041  1 ! ENVIRONMENT:  VAX/VMS user mode
;    42          0042  1 !
;    43          0043  1 ! AUTHOR: Peter D Gilbert, CREATION DATE: 17-Oct-1983
;    44          0044  1 !
;    45          0045  1 ! MODIFIED BY:
;    46          0046  1 !
;    47          0047  1 !       T03-001         Original
;    48          0048  1 !--
```

```
;    50          0049  1 LIBRARY 'SYS$LIBRARY:LIB';
;    51          0050  1 REQUIRE 'SRC$:SORMSG';
;    52          0227  1
;    53          0228  1 FORWARD ROUTINE
;    54          0229  1     !
;    55          0230  1     ! Global routines that interface to the CLI
;    56          0231  1     !
;    57          0232  1     CLI_BEGIN,                              ! Setup CLI processing
;    58          0233  1     CLI_GET_VALUE,                          ! Get value of item
;    59          0234  1     CLI_PRESENT,                            ! Determine if item is present
;    60          0235  1     CLI_NEXT_QUAL,                          ! Get next qualifier
;    61          0236  1     CLI_END,                                ! End CLI processing
;    62          0237  1     !
;    63          0238  1     ! Utility routines
;    64          0239  1     !
;    65          0240  1     INIT_MBX:           NOVALUE,            ! Initialize mailbox
;    66          0241  1     GET_SYS_LOG:        NOVALUE,            ! Translate system logical
;    67          0242  1     CLI_CHECKPOINT;                         ! Field requests
;    68          0243  1
;    69          0244  1 EXTERNAL ROUTINE
;    70          0245  1     STR$COPY_R:         ADDRESSING_MODE(GENERAL),
;    71          0246  1     CHK$CHKPNT:         ADDRESSING_MODE(GENERAL) WEAK,
;    72          0247  1     LIB$SYS_TRNLOG:     ADDRESSING_MODE(GENERAL),
;    73          0248  1     STR$COPY_R:         ADDRESSING_MODE(GENERAL);
;    74          0249  1
;    75          0250  1 EXTERNAL LITERAL
;    76          0251  1     CLI$_FACILITY,
;    77          0252  1     CLI$_SPAWNED,
;    78          0253  1     CLI$_INVREQTYP,
;    79          0254  1     CHK$_NOTINIT:       WEAK;
;    80          0255  1
;    81          0256  1 LITERAL FALSE = 0;
;    82          0257  1 LITERAL TRUE =  1;
;    83          0258  1 MACRO   BASE_ = 0,0,0,0 %;
;    84          0259  1 MACRO   ELIF =  ELSE IF %;
;    85          0260  1 MACRO   STR_MAILBOX =   'SORTMERGE_MBX' %;      ! Mailbox name
;    86          0261  1
;    87          0262  1
;    88          0263  1 ! Define the format of a message
;    89          0264  1 !
;    90          0265  1 FIELD
;    91          0266  1     MSG_FIELDS =
;    92          0267  1         SET
;    93          0268  1         MSG_L_ID=       [ 0,0,32,0],        ! Message ID
;    94          0269  1         MSG_L_STS=      [ 4,0,32,0],        ! Returned status of request
;    95          0270  1         MSG_L_RQST=     [ 8,0,32,0],        ! Request code
;    96          0271  1         MSG_L_LEN=      [12,0,32,0],        ! Length of following text
;    97          0272  1         MSG_A_TXT=      [16,0,32,0]         ! Start of text
;    98          0273  1         TES;
;    99          0274  1 LITERAL MSG_K_TEXT=     256;                        ! Largest text size
;   100          0275  1 LITERAL MSG_K_SIZE=     %FIELDEXPAND(MSG_A_TXT,0) + MSG_K_TEXT; ! Size in bytes
;   101          0276  1 MACRO   MSG_BLOCK=      BLOCK[MSG_K_SIZE,BYTE] FIELD(MSG_FIELDS) %;
;   102        L 0277  1 %MESSAGE('MSG_K_SIZE = ', %NUMBER(MSG_K_SIZE))
;   103          0278  1
;   104          0279  1 ! Values of MSG_L_ID
;   105          0280  1 !
;   106          0281  1 LITERAL
```

```
;  107          0282  1 !   MSG$_DELPROC          = whatever,
;  108          0283  1     MSG_K_CHKCLI          = MSG$_DELPROC + 1;
;  109          0284  1
;  110          0285  1 ! Values of MSG_L_RQST
;  111          0286  1 !
;  112          0287  1 LITERAL
;  113          0288  1     MSG_K_GET_VALUE       = 1,
;  114          0289  1     MSG_K_PRESENT         = 2,
;  115          0290  1     MSG_K_NEXT_QUAL       = 3,
;  116          0291  1     MSG_K_END             = 4;
;  117          0292  1
;  118          0293  1 OWN
;  119          0294  1     MBX_CHAN:   WORD,                   ! Mailbox channel
;  120          0295  1     MBX_UNIT,                           ! Mailbox unit
;  121          0296  1     CHK_SUBP:   BYTE,                   ! True if a checkpoint subprocess
;  122          0297  1     MSG:        MSG_BLOCK;              ! Message block
;  123          0298  1
;  124          0299  1 BIND
;  125          0300  1     CLI$_SHR_SYSERROR = SHR$_SYSERROR + STS$K_SEVERE + CLI$_FACILITY ^ 16,
;  126          0301  1     CLI$_SHR_BADLOGIC = SHR$_BADLOGIC + STS$K_SEVERE + CLI$_FACILITY ^ 16;
```

```
;  128          0302  1 GLOBAL ROUTINE CLI_BEGIN
;  129          0303  1          (
;  130          0304  1          IMAGE_AP:       REF $BBLOCK,     ! AP of main routine
;  131          0305  1          CHK_POINT                        ! True is checkpointing was specified
;  132          0306  1          ) =
;  133          0307  2      BEGIN
;  134          0308  2      LOCAL
;  135          0309  2          STATUS;
;  136          0310  2
;  137          0311  2      ! Determine whether we're a checkpoint subprocess
;  138          0312  2      !
;  139          0313  2      CHK_SUBP = FALSE;
;  140          0314  2 !    %IF %SWITCHES(DEBUG) %THEN
;  141          0315  2          IF .CHK_POINT EQL CLI$_INVREQTYP
;  142          0316  2          THEN
;  143          0317  2              CHK_SUBP = TRUE;
;  144          0318  2 !    %ELSE
;  145          0319  2 !        IF CHK$CHKPNT NEQ 0
;  146          0320  2 !        THEN
;  147          0321  2 !            BEGIN
;  148          0322  2 !            STATUS = CHK$CHKPNT();
;  149          0323  2 !            IF .STATUS AND .STATUS NEQ CHK$_NOTINIT THEN CHK_SUBP = TRUE;
;  150          0324  2 !            END;
;  151          0325  2 !    %FI
;  152          0326  2
;  153          0327  2      ! If we're a checkpoint subprocess, call the CLI through a mailbox
;  154          0328  2      !
;  155          0329  2      IF .CHK_SUBP
;  156          0330  2      THEN
;  157          0331  2          INIT_MBX()                   ! Attach to the communication mailbox
;  158          0332  2      ELIF .CHK_POINT
;  159          0333  2      THEN
;  160          0334  3          $EXIT(CODE = CLI_CHECKPOINT(IMAGE_AP[BASE_]))    ! Create subprocess
;  161          0335  2      ELSE
;  162          0336  2          0;                           ! Just continue normally
;  163          0337  2
;  164          0338  2      RETURN SS$_NORMAL;
;  165          0339  1      END;


                                        .TITLE  UTIL$CHK_CLI
                                        .IDENT  \V04-000\

                                        .PSECT  $OWN$,NOEXE,2

                                00000 MBX_CHAN:
                                        .BLKB   2
                                00002           .BLKB   2
                                00004 MBX_UNIT:
                                        .BLKB   4
                                00008 CHK_SUBP:
                                        .BLKB   1
                                00009           .BLKB   3
                                0000C MSG:      .BLKB   272

                                        .EXTRN  STR$COPY_R, LIB$SYS_TRNLOG
                                        .EXTRN  CLI$_FACILITY, CLI$_SPAWNED
```

```
                                                        .EXTRN  CLI$_INVREQTYP, SYS$EXIT
                                                        .WEAK   CHK$CHKPNT, CHK$_NOTINIT

                                                        .PSECT  $CODE$,NOWRT,2

                                        0004 00000      .ENTRY  CLI_BEGIN, Save R2                      ; 0302
                              52  0000' CF   9E 00002      MOVAB   CHK_SUBP, R2                        ;
                                         62   94 00007      CLRB    CHK_SUBP                            ; 0313
                    00000000G   8F   08  AC   D1 00009      CMPL    CHK_POINT, #CLI$_INVREQTYP          ; 0315
                                         03   12 00011      BNEQ    1$                                  ;
                              62         01   90 00013      MOVB    #1, CHK_SUBP                        ; 0317
                              07         62   E9 00016 1$:  BLBC    CHK_SUBP, 2$                        ; 0329
                         0000V  CF       00   FB 00019      CALLS   #0, INIT_MBX                        ; 0331
                                         15   11 0001E      BRB     3$                                  ;
                              11   08     AC   E9 00020 2$:  BLBC    CHK_POINT, 3$                       ; 0332
                                   04    AC   DD 00024      PUSHL   IMAGE_AP                            ; 0334
                         0000V  CF       01   FB 00027      CALLS   #1, CLI_CHECKPOINT                  ;
                                         50   DD 0002C      PUSHL   R0                                  ;
                    00000000G   00       01   FB 0002E      CALLS   #1, SYS$EXIT                        ;
                              50         01   D0 00035 3$:  MOVL    #1, R0                              ; 0338
                                         04   00038      RET                                         ; 0339

; Routine Size:  57 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  167          0340  1 GLOBAL ROUTINE CLI_END =
;  168          0341  2     BEGIN
;  169          0342  2     !+
;  170          0343  2     !
;  171          0344  2     ! This routine is called after all other calls to the CLI_xxx routines.
;  172          0345  2     ! It is called by a main process only if no checkpointing is requested.
;  173          0346  2     ! It is called by a checkpointed subprocess.
;  174          0347  2     !
;  175          0348  2     !-
;  176          0349  2     LOCAL
;  177          0350  2         STATUS;
;  178          0351  2
;  179          0352  2     IF NOT .CHK_SUBP THEN RETURN SS$_NORMAL;
;  180          0353  2
;  181          0354  2     ! Deassign the channel to the mailbox
;  182          0355  2     !
;  183          0356  2     STATUS = $DASSGN(CHAN=.MBX_CHAN);
;  184          0357  2     IF NOT .STATUS THEN RETURN SIGNAL(CLI$_SHR_SYERROR, 0, .STATUS);
;  185          0358  2
;  186          0359  2     RETURN SS$_NORMAL;
;  187          0360  1     END;


                                                        .EXTRN  SYS$DASSGN

                                                0000 00000      .ENTRY  CLI_END, Save nothing                           ; 0340
                                  21    0000'   CF   E9 00002  BLBC    CHK_SUBP, 1$                                    ; 0352
                                  7E    0000'   CF   3C 00007  MOVZWL  MBX_CHAN, -(SP)                                 ; 0356
                          00000000G   00        01   FB 0000C  CALLS   #1, SYS$DASSGN                                 ;
                                  12            50   E8 00013  BLBS    STATUS, 1$                                      ; 0357
                                                50   DD 00016  PUSHL   STATUS                                          ;
                                                7E   D4 00018  CLRL    -(SP)                                           ;
                                    00000000*   8F   DD 0001A  PUSHL   #<<CLI$_FACILITY@16>+4532>                      ;
                          00000000G   00        03   FB 00020  CALLS   #3, LIB$SIGNAL                                  ;
                                                04   00027  RET                                                        ;
                                  50            01   D0 00028  1$:     MOVL    #1, R0                                  ; 0359
                                                04   0002B  RET                                                        ; 0360

; Routine Size:  44 bytes,    Routine Base:  $CODE$ + 0039
```

```
;  189          0361  1 ! Read a message from a mailbox
;  190          0362  1 !
;  191          0363  1 MACRO
;  192        M 0364  1     MBX_GET (MSG_CODE, DESC) =
;  193        M 0365  1         BEGIN
;  194        M 0366  1         LOCAL S;
;  195        M 0367  1         LOCAL IOSB. VECTOR[4,BYTE];
;  196        M 0368  1         S = $QIOW(
;  197        M 0369  1             EFN=        0,
;  198        M 0370  1             CHAN=       .MBX_CHAN,
;  199        M 0371  1             FUNC=       IO$_READVBLK,
;  200        M 0372  1             IOSB=       IOSB[0],
;  201        M 0373  1             ASTADR=     0,
;  202        M 0374  1             ASTPRM=     0,
;  203        M 0375  1             P1= MSG[BASE_],
;  204        M 0376  1             P2= %ALLOCATION(MSG));
;  205        M 0377  1         IF .S THEN S = .IOSB[0];
;  206        M 0378  1         IF NOT .S THEN SIGNAL(CLI$_SHR_SYSERROR, 0, .S);
;  207        M 0379  1         %IF NOT %NULL(MSG_CODE) %THEN
;  208        M 0380  1             IF .MSG[MSG_L_ID] EQL MSG$_DELPROC
;  209        M 0381  1             THEN
;  210        M 0382  1                 $EXIT(CODE = .MSG[MSG_L_STS] OR STS$M_INHIB_MSG)
;  211        M 0383  1             ELIF
;  212        M 0384  1                 .MSG[MSG_L_ID] NEQ MSG_K_CHKCLI OR .MSG[MSG_L_RQST] NEQ MSG_CODE
;  213        M 0385  1             THEN
;  214        M 0386  1                 SIGNAL(CLI$_SHR_BADLOGIC);
;  215        M 0387  1         %FI
;  216        M 0388  1         %IF NOT %NULL(DESC) %THEN
;  217        M 0389  1             BEGIN
;  218        M 0390  1             STR$COPY_R(DESC, MSG[MSG_L_LEN], MSG[MSG_A_TXT]);
;  219        M 0391  1             END;
;  220        M 0392  1         %FI
;  221          0393  1         END %;
;  222          0394  1
;  223          0395  1 ! Write a message to a mailbox
;  224          0396  1 !
;  225          0397  1 MACRO
;  226        M 0398  1     MBX_PUT (MSG_CODE, DESC) =
;  227        M 0399  1         BEGIN
;  228        M 0400  1         LOCAL S;
;  229        M 0401  1         LOCAL IOSB: VECTOR[4,BYTE];
;  230        M 0402  1         %IF NOT %NULL(MSG_CODE) %THEN
;  231        M 0403  1             MSG[MSG_L_RQST] = MSG_CODE;
;  232        M 0404  1         %FI
;  233        M 0405  1         %IF NOT %NULL(DESC) %THEN
;  234        M 0406  1             CH$MOVE(MSG[MSG_L_LEN] = MINU(MSG_K_TEXT, .DESC[DSC$W_LENGTH]),
;  235        M 0407  1                 .DESC[DSC$A_POINTER], MSG[MSG_A_TXT]);
;  236        M 0408  1         %FI
;  237        M 0409  1         S = $QIOW(
;  238        M 0410  1             EFN=        0,
;  239        M 0411  1             CHAN=       .MBX_CHAN,
;  240        M 0412  1             FUNC=       IO$_WRITEVBLK,
;  241        M 0413  1             IOSB=       IOSB[0],
;  242        M 0414  1             ASTADR=     0,
;  243        M 0415  1             ASTPRM=     0,
;  244        M 0416  1             P1= MSG[BASE_],
;  245        M 0417  1             P2= (MSG[MSG_A_TXT] - MSG[BASE_]) + .MSG[MSG_L_LEN]);
```

```
;  246          M 0418  1          IF .S THEN S = .IOSB[0];
;  247          M 0419  1          IF NOT .S THEN SIGNAL(CLI$_SHR_SYSERROR, 0, .S);
;  248            0420  1          END %;
```

```
;  250          0421  1 ! These routines are used as an interface to the CLI.
;  251          0422  1 ! If sort is running as a checkpoint subprocess, there is no CLI.
;  252          0423  1 ! These routines allow us to pass information to the subprocess.
;  253          0424  1 !
;  254        M 0425  1 MACRO GEN_CLI_RTN_(RTN) =
;  255        M 0426  1     %NAME('CLI_',RTN)
;  256        M 0427  1         (
;  257        M 0428  1         NAME:   REF $BBLOCK
;  258        M 0429  1         %IF %IDENTICAL(RTN,'GET_VALUE') %THEN ,RETBUF: REF $BBLOCK %FI
;  259        M 0430  1         ) =
;  260        M 0431  1     BEGIN
;  261        M 0432  1     BUILTIN
;  262        M 0433  1         CALLG, AP;
;  263        M 0434  1     LOCAL
;  264        M 0435  1         STATUS;
;  265        M 0436  1
;  266        M 0437  1     ! Are we are a checkpoint subprocess?
;  267        M 0438  1     !
;  268        M 0439  1     IF .CHK_SUBP
;  269        M 0440  1     THEN
;  270        M 0441  1         BEGIN
;  271        M 0442  1         !
;  272        M 0443  1         ! Ask the main process for this.
;  273        M 0444  1         !
;  274        M 0445  1         MBX_PUT_(%NAME('MSG_K_',RTN), NAME);
;  275        M 0446  1         MBX_GET_(%NAME('MSG_K_',RTN),
;  276        M 0447  1             %IF %DECLARED(RETBUF) %THEN RETBUF[BASE_] %FI);
;  277        M 0448  1         STATUS = .MSG[MSG_L_STS];
;  278        M 0449  1         END
;  279        M 0450  1     ELSE
;  280        M 0451  1         BEGIN
;  281        M 0452  1         !
;  282        M 0453  1         ! Get the value from the CLI directly
;  283        M 0454  1         !
;  284        M 0455  1         EXTERNAL ROUTINE %NAME('CLI$',RTN): ADDRESSING_MODE(GENERAL);
;  285        M 0456  1         %IF %IDENTICAL(RTN,'NEXT_QUAL')          ! Hack for CLI$NEXT_QUAL
;  286        M 0457  1         %THEN
;  287        M 0458  1             OWN FIRST: BYTE INITIAL(TRUE);
;  288        M 0459  1             IF %NAME('CLI$',RTN) EQL 0
;  289        M 0460  1             THEN
;  290        M 0461  1                 BEGIN
;  291        M 0462  1                 STATUS = .FIRST;
;  292        M 0463  1                 FIRST = FALSE;
;  293        M 0464  1                 END
;  294        M 0465  1             ELSE
;  295        M 0466  1         %FI
;  296        M 0467  1         STATUS = CALLG(.AP, %NAME('CLI$',RTN));
;  297        M 0468  1         END;
;  298        M 0469  1     RETURN .STATUS;
;  299        M 0470  1     END %;
;  300          0471  1 GLOBAL ROUTINE GEN_CLI_RTN_('GET_VALUE');


                                                    .EXTRN  SYS$QIOW, CLI$GET_VALUE

                                01FC 00000          .ENTRY  CLI_GET_VALUE, Save R2,R3,R4,R5,R6,R7,R8    ; 0471
```

```
                                58 00000000G   00   9E 00002          MOVAB   SYS$QIOW, R8
                                57 00000000G   00   9E 00009          MOVAB   LIB$SIGNAL, R7
                                56     0000'   CF   9E 00010          MOVAB   MSG, R6
                                5E             08   C2 00015          SUBL2   #8, SP
                                03       FC    A6   E8 00018          BLBS    CHK_SUBP, 1$
                                               00C4  31 0001C          BRW     10$
                           08   A6             01   D0 0001F 1$:      MOVL    #1, MSG+8
                                50       04    AC   D0 00023          MOVL    NAME, R0
                                51             60   3C 00027          MOVZWL  (R0), R1
                          0100   8F             51   B1 0002A          CMPW    R1, #256
                                               05   1B 0002F          BLEQU   2$
                                51     0100    8F   3C 00031          MOVZWL  #256, R1
                           0C   A6             51   D0 00036 2$:      MOVL    R1, MSG+12
                10   A6     04   B0             51   28 0003A          MOVC3   R1, @4(R0), MSG+16
                                               7E   7C 00040          CLRQ    -(SP)
                                               7E   7C 00042          CLRQ    -(SP)
                     7E        0C   A6             10   C1 00044          ADDL3   #16, MSG+12, -(SP)
                                               56   DD 00049          PUSHL   R6
                                               7E   7C 0004B          CLRQ    -(SP)
                                     20        AE   9F 0004D          PUSHAB  IOSB
                                               30   DD 00050          PUSHL   #48
                                7E       F4    A6   3C 00052          MOVZWL  MBX_CHAN, -(SP)
                                               7E   D4 00056          CLRL    -(SP)
                                68             0C   FB 00058          CALLS   #12, SYS$QIOW
                                06             50   E9 0005B          BLBC    S, 3$
                                50             6E   9A 0005E          MOVZBL  IOSB, S
                                0D             50   E8 00061          BLBS    S, 4$
                                               50   DD 00064 3$:      PUSHL   S
                                               7E   D4 00066          CLRL    -(SP)
                                   00000000*   8F   DD 00068          PUSHL   #<<CLI$_FACILITY@16>+4532>
                                67             03   FB 0006E          CALLS   #3, LIB$SIGNAL
                                               7E   7C 00071 4$:      CLRQ    -(SP)
                                               7E   7C 00073          CLRQ    -(SP)
                                7E       0110  8F   3C 00075          MOVZWL  #272, -(SP)
                                               56   DD 0007A          PUSHL   R6
                                               7E   7C 0007C          CLRQ    -(SP)
                                     24        AE   9F 0007E          PUSHAB  IOSB
                                               31   DD 00081          PUSHL   #49
                                7E       F4    A6   3C 00083          MOVZWL  MBX_CHAN, -(SP)
                                               7E   D4 00087          CLRL    -(SP)
                                68             0C   FB 00089          CALLS   #12, SYS$QIOW
                                07             50   E9 0008C          BLBC    S, 5$
                                50       04    AE   9A 0008F          MOVZBL  IOSB, S
                                0D             50   E8 00093          BLBS    S, 6$
                                               50   DD 00096 5$:      PUSHL   S
                                               7E   D4 00098          CLRL    -(SP)
                                   00000000*   8F   DD 0009A          PUSHL   #<<CLI$_FACILITY@16>+4532>
                                67             03   FB 000A0          CALLS   #3, LIB$SIGNAL
                                03             66   D1 000A3 6$:      CMPL    MSG, #3
                                               12   12 000A6          BNEQ    7$
                     7E        04   A6 10000000  8F   C9 000A8          BISL3   #268435456, MSG+4, -(SP)
                          00000000G   00         01   FB 000B1          CALLS   #1, SYS$EXIT
                                               14   11 000B8          BRB     9$
                                04             66   D1 000BA 7$:      CMPL    MSG, #4
                                               06   12 000BD          BNEQ    8$
                                01       08    A6   D1 000BF          CMPL    MSG+8, #1
                                               09   13 000C3          BEQL    9$
```

```
                                00000000*  8F   DD 000C5 8$:     PUSHL   #<<CLI$_FACILITY@16>+4388>
                             67            01   FB 000CB         CALLS   #1, LIB$SIGNAL
                                     10    A6   9F 000CE 9$:     PUSHAB  MSG+16
                                     0C    A6   9F 000D1         PUSHAB  MSG+12
                                     08    AC   DD 000D4         PUSHL   RETBUF
                         00000000G   00          03   FB 000D7         CALLS   #3, STR$COPY_R
                                     50        04   A6   D0 000DE         MOVL    MSG+4, STATUS
                                               04   000E2         RET
                         00000000G   00          6C   FA 000E3 10$:     CALLG   (AP), CLI$GET_VALUE
                                               04   000EA         RET

; Routine Size:  235 bytes,    Routine Base:  $CODE$ + 0065

;  301          0472  1 GLOBAL ROUTINE GEN_CLI_RTN_('PRESENT');


                                                                  .EXTRN  CLI$PRESENT

                                                 01FC 00000       .ENTRY  CLI_PRESENT, Save R2,R3,R4,R5,R6,R7,R8      ; 0472
                             58 00000000G   00   9E   00002       MOVAB   SYS$QIOW, R8
                             57 00000000G   00   9E   00009       MOVAB   LIB$SIGNAL, R7
                             56     0000'   CF   9E   00010       MOVAB   MSG, R6
                             5E             08   C2   00015       SUBL2   #8, SP
                             03         FC  A6   E8   00018       BLBS    CHK_SUBP, 1$
                                        00B4 31   0001C           BRW     10$
                          08 A6             02   D0   0001F 1$:     MOVL    #2, MSG+8
                             50         04  AC   D0   00023       MOVL    NAME, R0
                             51             60   3C   00027       MOVZWL  (R0), R1
                          0100 8F           51   B1   0002A       CMPW    R1, #256
                                            05   1B   0002F       BLEQU   2$
                             51      0100   8F   3C   00031       MOVZWL  #256, R1
                          0C A6             51   D0   00036 2$:     MOVL    R1, MSG+12
                       10 A6      04  B0    51   28   0003A       MOVC3   R1, @4(R0), MSG+16
                                            7E   7C   00040       CLRQ    -(SP)
                                            7E   7C   00042       CLRQ    -(SP)
                             7E     0C  A6    10   C1   00044       ADDL3   #16, MSG+12, -(SP)
                                            56   DD   00049       PUSHL   R6
                                            7E   7C   0004B       CLRQ    -(SP)
                                      20    AE   9F   0004D       PUSHAB  IOSB
                                            30   DD   00050       PUSHL   #48
                                      7E    F4   A6   3C   00052       MOVZWL  MBX_CHAN, -(SP)
                                            7E   D4   00056       CLRL    -(SP)
                                      68         0C   FB   00058       CALLS   #12, SYS$QIOW
                                      06         50   E9   0005B       BLBC    S, 3$
                                      50         6E   9A   0005E       MOVZBL  IOSB, S
                                      0D         50   E8   00061       BLBS    S, 4$
                                                 50   DD   00064 3$:     PUSHL   S
                                                 7E   D4   00066       CLRL    -(SP)
                                00000000*        8F   DD   00068       PUSHL   #<<CLI$_FACILITY@16>+4532>
                             67                  03   FB   0006E       CALLS   #3, LIB$SIGNAL
                                                 7E   7C   00071 4$:     CLRQ    -(SP)
                                                 7E   7C   00073       CLRQ    -(SP)
                                      7E   0110   8F   3C   00075       MOVZWL  #272, -(SP)
                                                 56   DD   0007A       PUSHL   R6
                                                 7E   7C   0007C       CLRQ    -(SP)
```

```
                                 24    AE  9F 0007E          PUSHAB  IOSB
                                       31  DD 00081          PUSHL   #49
                              7E       F4  A6  3C 00083      MOVZWL  MBX_CHAN, -(SP)
                                           7E  D4 00087      CLRL    -(SP)
                              68           0C  FB 00089      CALLS   #12, SYS$QIOW
                              07           50  E9 0008C      BLBC    S, 5$
                              50       04  AE  9A 0008F      MOVZBL  IOSB, S
                              0D           50  E8 00093      BLBS    S, 6$
                                           50  DD 00096 5$:  PUSHL   S
                                           7E  D4 00098      CLRL    -(SP)
                                 00000000* 8F  DD 0009A      PUSHL   #<<CLI$_FACILITY@16>+4532>
                              67           03  FB 000A0      CALLS   #3, LIB$SIGNAL
                              03           66  D1 000A3 6$:  CMPL    MSG, #3
                                           12  12 000A6      BNEQ    7$
                        7E        04    A6 10000000  8F  C9 000A8  BISL3   #268435456, MSG+4, -(SP)
                           00000000G    00           01  FB 000B1  CALLS   #1, SYS$EXIT
                                           14  11 000B8      BRB     9$
                              04           66  D1 000BA 7$:  CMPL    MSG, #4
                                           06  12 000BD      BNEQ    8$
                              02       08  A6  D1 000BF      CMPL    MSG+8, #2
                                           09  13 000C3      BEQL    9$
                                 00000000* 8F  DD 000C5 8$:  PUSHL   #<<CLI$_FACILITY@16>+4388>
                              67           01  FB 000CB      CALLS   #1, LIB$SIGNAL
                              50       04  A6  D0 000CE 9$:  MOVL    MSG+4, STATUS
                                           04  000D2        RET
                           00000000G    00           6C  FA 000D3 10$: CALLG   (AP), CLI$PRESENT
                                           04  000DA        RET

; Routine Size:  219 bytes,    Routine Base:  $CODE$ + 0150

;  302          0473  1 GLOBAL ROUTINE GEN_CLI_RTN_('NEXT_QUAL');


                                                            .PSECT  $OWN$,NOEXE,2

                                        01  0011C FIRST:   .BYTE   1                                                       ;

                                                            .EXTRN  CLI$NEXT_QUAL

                                                            .PSECT  $CODE$,NOWRT,2

                                          03FC 00000        .ENTRY  CLI_NEXT_QUAL, Save R2,R3,R4,R5,R6,R7,R8,R9 ; 0473
                              59 00000000G  00   9E 00002   MOVAB   CLI$NEXT_QUAL, R9
                              58 00000000G  C0   9E 00009   MOVAB   SYS$QIOW, R8
                              57 00000000G  00   9E 00010   MOVAB   LIB$SIGNAL, R7
                              56     0000'  CF   9E 00017   MOVAB   MSG, R6
                              5E            08   C2 0001C   SUBL2   #8, SP
                              03       FC   A6   E8 0001F   BLBS    CHK_SUBP, 1$
                                      00B5  31   00023       BRW     10$
                        08    A6            03   D0 00026 1$:  MOVL    #3, MSG+8
                              50       04   AC   D0 0002A   MOVL    NAME, R0
                              51            60   3C 0002E   MOVZWL  (R0), R1
                        0100  8F            51   B1 00031   CMPW    R1, #256
                                            05   1B 00036   BLEQU   2$
                              51     0100   8F   3C 00038   MOVZWL  #256, R1
```

```
                    0C   A6          51  DO 0003D 2$:    MOVL    R1, MSG+12
                10  A6   04   B0     51  28 00041        MOVC3   R1, @4(R0), MSG+16
                                     7E  7C 00047        CLRQ    -(SP)
                                     7E  7C 00049        CLRQ    -(SP)
                    7E       0C   A6  10  C1 0004B        ADDL3   #16, MSG+12, -(SP)
                                     56  DD 00050        PUSHL   R6
                                     7E  7C 00052        CLRQ    -(SP)
                             20   AE  9F  00054        PUSHAB  IOSB
                                     30  DD 00057        PUSHL   #48
                         7E  F4   A6  3C 00059        MOVZWL  MBX_CHAN, -(SP)
                                     7E  D4 0005D        CLRL    -(SP)
                         68          0C  FB 0005F        CALLS   #12, SYS$QIOW
                         06          50  E9 00062        BLBC    S, 3$
                         50          6E  9A 00065        MOVZBL  IOSB, S
                         0D          50  E8 00068        BLBS    S, 4$
                                     50  DD 0006B 3$:    PUSHL   S
                                     7E  D4 0006D        CLRL    -(SP)
                    C0000000*        8F  DD 0006F        PUSHL   #<<CLI$_FACILITY@16>+4532>
                         67          03  FB 00075        CALLS   #3, LIB$SIGNAL
                                     7E  7C 00078 4$:    CLRQ    -(SP)
                                     7E  7C 0007A        CLRQ    -(SP)
                         7E    0110  8F  3C 0007C        MOVZWL  #272, -(SP)
                                     56  DD 00081        PUSHL   R6
                                     7E  7C 00083        CLRQ    -(SP)
                             24   AE  9F  00085        PUSHAB  IOSB
                                     31  DD 00088        PUSHL   #49
                         7E  F4   A6  3C 0008A        MOVZWL  MBX_CHAN, -(SP)
                                     7E  D4 0008E        CLRL    -(SP)
                         68          0C  FB 00090        CALLS   #12, SYS$QIOW
                         07          50  E9 00093        BLBC    S, 5$
                         50    04   AE  9A 00096        MOVZBL  IOSB, S
                         0D          50  E8 0009A        BLBS    S, 6$
                                     50  DD 0009D 5$:    PUSHL   S
                                     7E  D4 0009F        CLRL    -(SP)
                    00000000*        8F  DD 000A1        PUSHL   #<<CLI$_FACILITY@16>+4532>
                         67          03  FB 000A7        CALLS   #3, LIB$SIGNAL
                         03          66  D1 000AA 6$:    CMPL    MSG, #3
                                     12  12 000AD        BNEQ    7$
7E        04   A6  10000000          8F  C9 000AF        BISL3   #268435456, MSG+4, -(SP)
    00000000G    00                 01  FB 000B8        CALLS   #1, SYS$EXIT
                                     14  11 000BF        BRB     9$
                         04          66  D1 000C1 7$:    CMPL    MSG, #4
                                     06  12 000C4        BNEQ    8$
                         03    08   A6  D1 000C6        CMPL    MSG+8, #3
                                     09  13 000CA        BEQL    9$
                    00000000*        8F  DD 000CC 8$:    PUSHL   #<<CLI$_FACILITY@16>+4388>
                         67          01  FB 000D2        CALLS   #1, LIB$SIGNAL
                         52    04   A6  D0 000D5 9$:    MOVL    MSG+4, STATUS
                                     16  11 000D9        BRB     12$
                         50          69  9E 000DB 10$:   MOVAB   CLI$NEXT_QUAL, R0
                                     0B  12 000DE        BNEQ    11$
                         52    0110  C6  9A 000E0        MOVZBL  FIRST, STATUS
                               0110  C6  94 000E5        CLRB    FIRST
                                     06  11 000E9        BRB     12$
                         69          6C  FA 000EB 11$:   CALLG   (AP), CLI$NEXT_QUAL
                         52          50  D0 000EE        MOVL    R0, STATUS
                         50          52  D0 000F1 12$:   MOVL    STATUS, R0
```

```
                                04 000F4        RET                                             ;

; Routine Size:  245 bytes,    Routine Base:  $CODE$ + 022B
```

```
;   304          0474  1 KEYWORDMACRO
;   305          0475  1     $CRECHK(
;   306          0476  1         PIDADR=0, IMAGE=0, INPUT=0, OUTPUT=0, ERROR=0, PRVADR=0,
;   307          0477  1         QUOTA=0, PRCNAM=0, BASPRI=0, UIC=0, MBXUNT=0, STSFLG=0,
;   308        M 0478  1         FLAGS=0, RSTMODE=0, EXTSTA=0, PAGFIL=0) =
;   309        M 0479  1             BEGIN
;   310        M 0480  1             MACRO
;   311        M 0481  1                 F_(X) =
;   312        M 0482  1                     %IF %CTCE(X)
;   313        M 0483  1                     %THEN
;   314        M 0484  1                         %IF X NEQ 0 %THEN %ERROR('Error in $CRECHK') %FI
;   315        M 0485  1                         X
;   316        M 0486  1                     %ELSE
;   317        M 0487  1                         (IF CHK$CRECHK NEQ 0 THEN X ELSE .X)
;   318        M 0488  1                     %FI %QUOTE %;
;   319        M 0489  1             EXTERNAL ROUTINE
;   320        M 0490  1                 SYS$CREPRC: ADDRESSING_MODE(GENERAL),
;   321        M 0491  1                 CHK$CRECHK: ADDRESSING_MODE(GENERAL) WEAK;
;   322        M 0492  1             (IF CHK$CRECHK NEQ 0 THEN CHK$CRECHK ELSE SYS$CREPRC)
;   323        M 0493  1                 (PIDADR, IMAGE, INPUT, OUTPUT, ERROR, PRVADR,
;   324        M 0494  1                  QUOTA, PRCNAM, F_(BASPRI), F_(UIC), F_(MBXUNT), F_(STSFLG)
;   325        M 0495  1                  ,F_(FLAGS), RSTMODE, EXTSTA, PAGFIL
;   326        M 0496  1                 )
;   327          0497  1             END %;
```

```
;  329          0498  1 ROUTINE CLI_CHECKPOINT(
;  330          0499  1          IMAGE_AP:       REF $BBLOCK
;  331          0500  1          ) =
;  332          0501  2      BEGIN
;  333          0502  2      !
;  334          0503  2      ! Create a checkpoint subprocess and field it's requests.
;  335          0504  2      !
;  336          0505  2      MACRO
;  337          0506  2          STR_INPUT =     'SYS$INPUT' %,
;  338          0507  2          STR_OUTPUT =    'SYS$OUTPUT' %,
;  339          0508  2          STR_ERROR =     'SYS$ERROR' %;
;  340          0509  2      MACRO
;  341        M 0510  2          DYNAMIC_STRING = $BBLOCK[DSC$C_D_BLN]
;  342        M 0511  2              PRESET(
;  343        M 0512  2                  [DSC$W_LENGTH] = 0,
;  344        M 0513  2                  [DSC$B_CLASS] = DSC$K_CLASS_D,
;  345        M 0514  2                  [DSC$B_DTYPE] = DSC$K_DTYPE_T,
;  346          0515  2                  [DSC$A_POINTER] = 0) %;
;  347          0516  2      LOCAL
;  348          0517  2          BUFFER: $BBLOCK[64];
;  349          0518  2      LOCAL
;  350          0519  2          DYN:    DYNAMIC_STRING,
;  351          0520  2          INPUT:  DYNAMIC_STRING,
;  352          0521  2          OUTPUT: DYNAMIC_STRING,
;  353          0522  2          ERROR:  DYNAMIC_STRING,
;  354          0523  2          PRCNAM: $BBLOCK[DSC$C_S_BLN],
;  355          0524  2          STATUS;
;  356          0525  2
;  357          0526  2      ! First create the communication mailbox
;  358          0527  2      !
;  359          0528  2      INIT_MBX();
;  360          0529  2
;  361          0530  2      ! Get the names of system input/output/error logicals
;  362          0531  2      !
;  363          0532  3      BEGIN
;  364          0533  3      MACRO
;  365        M 0534  3          LENADR_R_(X) = %IF %ISSTRING(X)
;  366        M 0535  3              %THEN %REF(%CHARCOUNT(X)), UPLIT BYTE(X)
;  367          0536  3              %ELSE X[DSC$W_LENGTH], .X[DSC$A_POINTER] %FI %;
;  368          0537  3      STR$COPY_R(INPUT [BASE_], LENADR_R_(STR_INPUT ));
;  369          0538  3      STR$COPY_R(OUTPUT[BASE_], LENADR_R_(STR_OUTPUT));
;  370          0539  3      STR$COPY_R(ERROR [BASE_], LENADR_R_(STR_ERROR ));
;  371          0540  3      GET_SYS_LOG(INPUT [BASE_]);
;  372          0541  3      GET_SYS_LOG(OUTPUT[BASE_]);
;  373          0542  3      GET_SYS_LOG(ERROR [BASE_]);
;  374          0543  2      END;
;  375          0544  2
;  376          0545  2      ! Kickoff subprocess
;  377          0546  2      !
;  378          0547  2      INCR I FROM 1 TO 255 DO
;  379          0548  3          BEGIN
;  380          0549  3          LOCAL PID, BASPRI, FLAGS;
;  381          0550  3          LOCAL PRCJPI: $BBLOCK[8] INITIAL(%ALLOCATION(BUFFER), BUFFER[BASE_]);
;  382          0551  3          LOCAL ITMLST: VECTOR[3+3+1] INITIAL(
;  383          0552  3              WORD(%ALLOCATION(BUFFER), JPI$_USERNAME), BUFFER[BASE_], PRCJPI[DSC$W_LENGTH],
;  384          0553  3              WORD(%ALLOCATION(BASPRI), JPI$_PRIB), BASPRI, 0,
;  385          0554  3              0);
```

```
;  386          0555  3          LOCAL P;
;  387        P 0556  3          STATUS = $GETJPIW(
;  388        P 0557  3                  EFN = 0,
;  389          0558  3                  ITMLST = ITMLST[0]);
;  390          0559  3          P = CH$FIND_CH(.PRCJPI[DSC$W_LENGTH], .PRCJPI[DSC$A_POINTER], %C' ');
;  391          0560  3          IF CH$FAIL(.P) THEN P = .PRCJPI[DSC$A_POINTER] + .PRCJPI[DSC$W_LENGTH];
;  392          0561  3          PRCJPI[DSC$W_LENGTH] = .P - .PRCJPI[DSC$A_POINTER];
;  393          0562  3          PRCNAM[DSC$W_LENGTH] = %ALLOCATION(BUFFER);
;  394          0563  3          PRCNAM[DSC$B_DTYPE] = 0;
;  395          0564  3          PRCNAM[DSC$B_CLASS] = 0;
;  396          0565  3          PRCNAM[DSC$A_POINTER] = BUFFER[BASE_];
;  397        P 0566  3          STATUS = $FAO($DESCRIPTOR('!AS_!UL'),
;  398          0567  3                  PRCNAM[DSC$W_LENGTH], PRCNAM[BASE_], PRCJPI[BASE_], .I);
;  399          0568  3          IF NOT .STATUS THEN RETURN SIGNAL(CLI$_SHR_SYSERROR, 0, .STATUS);
;  400          0569  3          FLAGS = CHKPNT$M_AUTO_RESTART OR CHKPNT$M_ASYNC;
;  401        P 0570  3          STATUS = $CRECHK(
;  402        P 0571  3              PIDADR=     PID,
;  403        P 0572  3              IMAGE=
;  404        P 0573  3 !            %IF %SWITCHES(DEBUG) %THEN
;  405        P 0574  3 !                    $BBLOCK[.IMAGE_AP[CLI$A_IMGFILED],IFD$Q_CURPROG],
;  406        P 0575  3 !            %ELSE
;  407        P 0576  3 !                    (LOCAL D: VECTOR[2]; D[1] =
;  408        P 0577  3 !                    $BBLOCK[.IMAGE_AP[CLI$A_IMGHDADR],IHX$T_IMGNAM];
;  409        P 0578  3 !                    D[0] = CH$RCHAR_A(D[1]); D[0]),
;  410        P 0579  3 !            %FI
;  411        P 0580  3              INPUT=      INPUT [BASE_],
;  412        P 0581  3              OUTPUT=     OUTPUT[BASE_],
;  413        P 0582  3              ERROR=      ERROR [BASE_],
;  414        P 0583  3              PRCNAM=     PRCNAM[BASE_],
;  415        P 0584  3              BASPRI=     BASPRI,
;  416        P 0585  3              FLAGS=      FLAGS,
;  417          0586  3              MBXUNT=     MBX_UNIT);
;  418          0587  3          IF .STATUS NEQ SS$_DUPLNAM THEN EXITLOOP;
;  419          0588  2          END;
;  420          0589  2      IF NOT .STATUS THEN RETURN SIGNAL(CLI$_SHR_SYSERROR, 0, .STATUS);
;  421          0590  2      SIGNAL(CLI$_SPAWNED, 1, PRCNAM[BASE_]);
;  422          0591  2
;  423          0592  2
;  424          0593  2      WHILE TRUE DO
;  425          0594  3          BEGIN
;  426          0595  3          LOCAL
;  427          0596  3              D:  VECTOR[2];
;  428          0597  3
;  429          0598  3          ! Read a request
;  430          0599  3          !
;  431          0600  3          MBX_GET (,);
;  432          0601  3          IF .MSG[MSG_L_ID] EQL MSG$_DELPROC
;  433          0602  3          THEN
;  434          0603  3              RETURN .MSG[MSG_L_STS] OR STS$M_INHIB_MSG
;  435          0604  3          ELIF
;  436          0605  3              .MSG[MSG_L_ID] NEQ MSG_K_CHKCLI
;  437          0606  3          THEN
;  438          0607  3              SIGNAL(CLI$_SHR_BADLOGIC);
;  439          0608  3
;  440          0609  3          ! Field the request
;  441          0610  3          !
;  442          0611  3          D[0] = .MSG[MSG_L_LEN];
```

```
;  443          0612  3          D[1] =  MSG[MSG_A_TXT];
;  444          0613  3          CASE .MSG[MSG_L_RQST] FROM 1 TO 3 OF
;  445          0614  3              SET
;  446          0615  3              [MSG_K_GET_VALUE]:
;  447          0616  4                  BEGIN
;  448          0617  4                  MSG[MSG_L_STS] = CLI_GET_VALUE(D[0], DYN[BASE_]);
;  449          0618  4                  CH$MOVE(MSG[MSG_L_LEN] = MINU(MSG_K_TEXT, .DYN[DSC$W_LENGTH]),
;  450          0619  4                      .DYN[DSC$A_POINTER], MSG[MSG_A_TXT]);
;  451          0620  3                  END;
;  452          0621  3              [MSG_K_PRESENT]:
;  453          0622  4                  BEGIN
;  454          0623  4                  MSG[MSG_L_STS] = CLI_PRESENT(D[0]);
;  455          0624  4                  MSG[MSG_L_LEN] = 0;
;  456          0625  3                  END;
;  457          0626  3              [MSG_K_NEXT_QUAL]:
;  458          0627  4                  BEGIN
;  459          0628  4                  MSG[MSG_L_STS] = CLI_NEXT_QUAL(D[0]);
;  460          0629  4                  MSG[MSG_L_LEN] = 0;
;  461          0630  3                  END;
;  462          0631  3              [OUTRANGE]:
;  463          0632  3                  RETURN SIGNAL(CLI$_SHR_BADLOGIC);
;  464          0633  3              TES;
;  465          0634  3
;  466          0635  3          ! Send the response
;  467          0636  3          !
;  468          0637  3          MBX_PUT_(,);
;  469          0638  2          END;
;  470          0639  2
;  471          0640  2      RETURN SS$_NORMAL;
;  472          0641  1      END;
; INFO#250                 L1:0586
; Referenced LOCAL symbol BASPRI is probably not initialized


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                    54  55  50  4E  49  24  53  59  53  00000 P.AAA:  .ASCII  \SYS$INPUT\
                54  55  50  54  55  4F  24  53  59  53  00009 P.AAB:  .ASCII  \SYS$OUTPUT\
                    52  4F  52  52  45  24  53  59  53  00013 P.AAC:  .ASCII  \SYS$ERROR\
                                        0202  0040  0001C P.AAD:  .WORD   64, 514
                                          00000000  00020          .LONG   0
                                          00000000  00024          .LONG   0
                                        0309  0004  00028          .WORD   4, 777
                            00000000  00000000  00000000  0002C          .LONG   0, 0, 0
                              4C  55  21  5F  53  41  21  00038 P.AAF:  .ASCII  \!AS_!UL\
                                                  0003F          .BLKB   1
                                          00000007  00040 P.AAE:  .LONG   7
                                          00000000' 00044          .ADDRESS P.AAF

                                                        .EXTRN  SYS$GETJPIW, SYS$FAO
                                                        .EXTRN  SYS$CREPRC
                                                        .WEAK   CHK$CRECHK

                                                        .PSECT  $CODE$,NOWRT,2

                                                OFFC 00000 CLI_CHECKPOINT:
```

```
                                                .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11    ; 0498
                5B 00000000G   00    9E 00002   MOVAB   STR$COPY_R, R11
                5A 00000000G   00    9E 00009   MOVAB   LIB$SIGNAL, R10
                59      0000'  CF    9E 00010   MOVAB   MSG, R9
                5E      FF5C   CE    9E 00015   MOVAB   -164(SP), SP
           5C   AE 020E0000    8F    D0 0001A   MOVL    #34471936, DYN                          ; 0519
                        60     AE    D4 00022   CLRL    DYN+4
           54   AE 020E0000    8F    D0 00025   MOVL    #34471936, INPUT                        ; 0520
                        58     AE    D4 0002D   CLRL    INPUT+4
           4C   AE 020E0000    8F    D0 00030   MOVL    #34471936, OUTPUT                       ; 0521
                        50     AE    D4 00038   CLRL    OUTPUT+4
           44   AE 020E0000    8F    D0 0003B   MOVL    #34471936, ERROR                        ; 0522
                        48     AE    D4 00043   CLRL    ERROR+4
      0000V     CF             00    FB 00046   CALLS   #0, INIT_MBX                           ; 0528
                      0000'    CF    9F 0004B   PUSHAB  P.AAA                                   ; 0537
           04   AE             09    D0 0004F   MOVL    #9, 4(SP)
                        04     AE    9F 00053   PUSHAB  4(SP)
                        5C     AE    9F 00056   PUSHAB  INPUT
                   6B           03    FB 00059   CALLS   #3, STR$COPY_R
                      0000'    CF    9F 0005C   PUSHAB  P.AAB                                   ; 0538
           04   AE             0A    D0 00060   MOVL    #10, 4(SP)
                        04     AE    9F 00064   PUSHAB  4(SP)
                        54     AE    9F 00067   PUSHAB  OUTPUT
                   6B           03    FB 0006A   CALLS   #3, STR$COPY_R
                      0000'    CF    9F 0006D   PUSHAB  P.AAC                                   ; 0539
           04   AE             09    D0 00071   MOVL    #9, 4(SP)
                        04     AE    9F 00075   PUSHAB  4(SP)
                        4C     AE    9F 00078   PUSHAB  ERROR
                   6B           03    FB 0007B   CALLS   #3, STR$COPY_R
                        54     AE    9F 0007E   PUSHAB  INPUT                                   ; 0540
      0000V     CF             01    FB 00081   CALLS   #1, GET_SYS_LOG
                        4C     AE    9F 00086   PUSHAB  OUTPUT                                  ; 0541
      0000V     CF             01    FB 00089   CALLS   #1, GET_SYS_LOG
                        44     AE    9F 0008E   PUSHAB  ERROR                                   ; 0542
      0000V     CF             01    FB 00091   CALLS   #1, GET_SYS_LOG
                        56     04    AC D0 00096   MOVL    IMAGE_AP, R6                        ; 0586
                        58     01    D0 0009A   MOVL    #1, I
                  34   AE      40    8F 9A 0009D 1$:  MOVZBL  #64, PRCJPI                      ; 0550
                  38   AE      64    AE 9E 000A2   MOVAB   BUFFER, PRCJPI+4
           18 AE       0000'  CF    1C 28 000A7   MOVC3   #28, P.AAD, ITMLST                   ; 0554
                  1C   AE      64    AE 9E 000AE   MOVAB   BUFFER, ITMLST+4                     ; 0552
                  20   AE      34    AE 9E 000B3   MOVAB   PRCJPI, ITMLST+8
                  28   AE      08    AE 9E 000B8   MOVAB   BASPRI, ITMLST+16
                               7E    7C 000BD   CLRQ    -(SP)                                   ; 0558
                               7E    D4 000BF   CLRL    -(SP)
                               24    AE 9F 000C1   PUSHAB  ITMLST
                               7E    7C 000C4   CLRQ    -(SP)
                               7E    D4 000C6   CLRL    -(SP)
                00000000G      00    07 FB 000C8   CALLS   #7, SYS$GETJPIW
                               57    50 D0 000CF   MOVL    R0, STATUS
           38 BE       34   AE      20 3A 000D2   LOCC    #32, PRCJPI, @PRCJPI+4              ; 0559
                               02    12 000D8   BNEQ    2$
                               51    D4 000DA   CLRL    R1
                               51    D5 000DC 2$:  TSTL    P                                   ; 0560
                               08    12 000DE   BNEQ    3$
                         51    34   AE 3C 000E0   MOVZWL  PRCJPI, P
                         51    38   AE C0 000E4   ADDL2   PRCJPI+4, P
```

```
                34   AE                 51        38   AE   A3 000E8 3$:     SUBW3   PRCJPI+4, P, PRCJPI                 ; 0561
                                     3C   AE        40   8F   9A 000EE         MOVZBL  #64, PRCNAM                 ; 0562
                                     40   AE        64   AE   9E 000F3         MOVAB   BUFFER, PRCNAM+4            ; 0565
                                                    58   DD   00F8            PUSHL   I                           ; 0567
                                                    38   AE   9F 000FA         PUSHAB  PRCJPI
                                                    44   AE   9F 000FD         PUSHAB  PRCNAM
                                                    48   AE   9F 00100         PUSHAB  PRCNAM
                                                  0000'  CF   9F 00103         PUSHAB  P.AAE
                                    00000000G     00        05   FB 00107     CALLS   #5, SYS$FAO
                                                  57        50   D0 0010E     MOVL    R0, STATUS
                                                  03        57   E8 00111     BLBS    STATUS, 4$                  ; 0568
                                                          0087   31 00114     BRW     14$
                                     04   AE                05   D0 00117 4$: MOVL    #5, FLAGS                   ; 0569
                                                  50 00000000G  00   9E 0011B MOVAB   CHK$CRECHK, R0               ; 0586
                                                                52   D4 00122 CLRL    R2
                                                                50   D5 00124 TSTL    R0
                                                                0B   13 00126 BEQL    5$
                                                                52   D6 00128 INCL    R2
                                                  50 00000000G  00   9E 0012A MOVAB   CHK$CRECHK, R0
                                                                07   11 00131 BRB     6$
                                                  50 00000000G  00   9E 00133 5$:  MOVAB   SYS$CREPRC, R0
                                                                7E   7C 0013A 6$:  CLRQ    -(SP)
                                                                7E   D4 0013C     CLRL    -(SP)
                                                  08            52   E9 0013E BLBC    R2, 7$
                                                  51        10  AE   9E 00141 MOVAB   FLAGS, R1
                                                                51   DD 00145 PUSHL   R1
                                                                03   11 00147 BRB     8$
                                                            10  AE   DD 00149 7$:  PUSHL   FLAGS
                                                                7E   D4 0014C 8$:  CLRL    -(SP)
                                                  08            52   E9 0014E BLBC    R2, 9$
                                                  51        F8  A9   9E 00151 MOVAB   MBX_UNIT, R1
                                                                51   DD 00155 PUSHL   R1
                                                                03   11 00157 BRB     10$
                                                            F8  A9   DD 00159 9$:  PUSHL   MBX_UNIT
                                                                7E   D4 0015C 10$: CLRL    -(SP)
                                                  08            52   E9 0015E BLBC    R2, 11$
                                                  51        24  AE   9E 00161 MOVAB   BASPRI, R1
                                                                51   DD 00165 PUSHL   R1
                                                                03   11 00167 BRB     12$
                                                            24  AE   DD 00169 11$: PUSHL   BASPRI
                                                            5C  AE   9F 0016C 12$: PUSHAB  PRCNAM
                                                                7E   7C 0016F     CLRQ    -(SP)
                                                            70  AE   9F 00171     PUSHAB  ERROR
                                                            7C  AE   9F 00174     PUSHAB  OUTPUT
                                                            B0  AD   9F 00177     PUSHAB  INPUT
                                          7E        10  A6        14   C1 0017A ADDL3   #20, 16(R6), -(SP)
                                                        48  AE   9F 0017F     PUSHAB  PID
                                                  60            10   FB 00182 CALLS   #16, (R0)
                                                  57            50   D0 00185 MOVL    R0, STATUS
                                        00000094  8F            57   D1 00188 CMPL    STATUS, #148                ; 0587
                                                                0A   12 0018F BNEQ    13$
                        FF02        58            01 000000FF  8F   F1 00191 ACBL    #255, #1, I, 1$             ; 0547
                                                  0E            57   E8 0019B 13$: BLBS    STATUS, 15$             ; 0589
                                                                57   DD 0019E 14$: PUSHL   STATUS
                                                                7E   D4 001A0     CLRL    -(SP)
                                                     00000000*  8F   DD 001A2     PUSHL   #<<CLI$_FACILITY@16>+4532>
                                                  6A            03   FB 001A8     CALLS   #3, LIB$SIGNAL
```

```
                                    04 001AB          RET
                              3C    AE 9F 001AC 15$:  PUSHAB  PRCNAM                                    ;0590
                                    01 DD 001AF       PUSHL   #1
                         00000000G  8F DD 001B1       PUSHL   #CLI$_SPAWNED
                              6A    03 FB 001B7 16$:  CALLS   #3, LIB$SIGNAL
                                    7E 7C 001BA 17$:  CLRQ    -(SP)                                     ;0600
                                    7E 7C 001BC       CLRQ    -(SP)
                              7E  0110  8F 3C 001BE   MOVZWL  #272, -(SP)
                                    59 DD 001C3       PUSHL   R9
                                    7E 7C 001C5       CLRQ    -(SP)
                              30    AE 9F 001C7       PUSHAB  IOSB
                                    31 DD 001CA       PUSHL   #49
                              7E    F4 A9 3C 001CC    MOVZWL  MBX_CHAN, -(SP)
                                    7E D4 001D0       CLRL    -(SP)
                   00000000G  00       0C FB 001D2    CALLS   #12, SYS$QIOW
                              07       50 E9 001D9    BLBC    S, 18$
                              50    10  AE 9A 001DC    MOVZBL  IOSB, S
                              0D       50 E8 001E0    BLBS    S, 19$
                                       50 DD 001E3 18$:  PUSHL   S
                                       7E D4 001E5    CLRL    -(SP)
                         00000000*     8F DD 001E7    PUSHL   #<<CLI$_FACILITY@16>+4532>
                              6A       03 FB 001ED    CALLS   #3, LIB$SIGNAL
                              03       69 D1 001F0 19$:  CMPL    MSG, #3                                 ;0601
                                       0A 12 001F3    BNEQ    20$
                50    04    A9 10000000  8F C9 001F5    BISL3   #268435456, MSG+4, R0                   ;0603
                                       04 001FE       RET
                              04       69 D1 001FF 20$:  CMPL    MSG, #4                                 ;0605
                                       09 13 00202    BEQL    21$
                         00000000*     8F DD 00204    PUSHL   #<<CLI$_FACILITY@16>+4388>               ;0607
                              6A       01 FB 0020A    CALLS   #1, LIB$SIGNAL
                        34    AE    0C  A9 D0 0020D 21$:  MOVL    MSG+12, D                           ;0611
                        38    AE    10  A9 9E 00212    MOVAB   MSG+16, D+4                             ;0612
                    02        01    08  A9 CF 00217    CASEL   MSG+8, #1, #2                           ;0613
                0045      003B      0010     0021C 22$:  .WORD   23$-22$,-
                                                         25$-22$,-
                                                         26$-22$
                         00000000*     8F DD 00222    PUSHL   #<<CLI$_FACILITY@16>+4388>               ;0632
                              6A       01 FB 00228    CALLS   #1, LIB$SIGNAL
                                       04 0022B       RET
                              5C    AE 9F 0022C 23$:  PUSHAB  DYN                                       ;0617
                              38    AE 9F 0022F       PUSHAB  D
                         FB0E CF       02 FB 00232    CALLS   #2, CLI_GET_VALUE
                           04 A9       50 D0 00237    MOVL    R0, MSG+4
                              50    5C  AE 3C 0023B    MOVZWL  DYN, R0                                 ;0618
                         0100 8F       50 B1 0023F    CMPW    R0, #256
                                       05 1B 00244    BLEQU   24$
                              50  0100  8F 3C 00246    MOVZWL  #256, R0
                           0C A9       50 D0 0024B 24$:  MOVL    R0, MSG+12
                10    A9    60 BE       50 28 0024F    MOVC3   R0, @DYN+4, MSG+16                      ;0619
                                       19 11 00255    BRB     28$                                     ;0613
                              34    AE 9F 00257 25$:  PUSHAB  D                                       ;0623
                         FBD1 CF       01 FB 0025A    CALLS   #1, CLI_PRESENT
                                       08 11 0025F    BRB     27$
                              34    AE 9F 00261 26$:  PUSHAB  D                                       ;0628
                         FCA2 CF       01 FB 00264    CALLS   #1, CLI_NEXT_QUAL
                           04 A9       50 D0 00269 27$:  MOVL    R0, MSG+4
                              0C    A9 D4 0026D       CLRL    MSG+12                                  ;0629
```

```
                                                7E  7C 00270 28$:     CLRQ    -(SP)                                   ; 0637
                                                7E  7C 00272        CLRQ    -(SP)                                   ;
                        7E      0C      A9      10  C1 00274        ADDL3   #16, MSG+12, -(SP)                      ;
                                                59  DD 00279        PUSHL   R9                                      ;
                                                7E  7C 0027B        CLRQ    -(SP)                                   ;
                                        34      AE  9F 0027D        PUSHAB  IOSB                                    ;
                                                30  DD 00280        PUSHL   #48                                     ;
                                7E      F4      A9  3C 00282        MOVZWL  MBX_CHAN, -(SP)                         ;
                                                7E  D4 00286        CLRL    -(SP)                                   ;
                            00000000G   00              0C  FB 00288        CALLS   #12, SYS$QIOW                   ;
                                        0A              50  E9 0028F        BLBC    S, 29$                          ;
                                        50      14      AE  9A 00292        MOVZBL  IOSB, S                         ;
                                        03              50  E9 00296        BLBC    S, 29$                          ;
                                                    FF1E    31 00299        BRW     17$                             ;
                                                50  DD 0029C 29$:     PUSHL   S                                       ;
                                                7E  D4 0029E        CLRL    -(SP)                                   ;
                                    00000000*   8F  DD 002A0        PUSHL   #<<CLI$_FACILITY@16>+4532>              ;
                                                FF0E    31 002A6        BRW     16$                                     ;

; Routine Size:  681 bytes,    Routine Base:  $CODE$ + 0320
```

```
Va
--
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
7F
7F
7F
7F
7F
7F
7F
7F
7F
7F
7F
7F
```

```
;  474          0642  1 ROUTINE GET_SYS_LOG(
;  475          0643  1          DSC:    REF $BBLOCK[DSC$C_D_BLN]
;  476          0644  1          ): NOVALUE =
;  477          0645  2      BEGIN
;  478          0646  2      LOCAL
;  479          0647  2          STATUS;
;  480          0648  2      STATUS = LIB$SYS_TRNLOG(DSC[BASE_], 0, DSC[BASE_]);
;  481          0649  2      IF NOT .STATUS THEN RETURN SIGNAL(CLI$_SHR_SYSERROR, 0, .STATUS);
;  482          0650  2      IF CH$RCHAR(.DSC[DSC$A_POINTER]) EQL %X'1B'          ! Escape?
;  483          0651  2      THEN
;  484          0652  2          STR$COPY_R(DSC[BASE_],
;  485          0653  2              %REF(.DSC[DSC$W_LENGTH]-4), .DSC[DSC$A_POINTER]+4);
;  486          0654  1      END;


                                0004 00000 GET_SYS_LOG:
                                           .WORD   Save R2                                  ; 0642
                         5E             04  C2 00002          SUBL2   #4, SP
                         52        04   AC  D0 00005          MOVL    DSC, R2                 ; 0648
                                        52  DD 00009          PUSHL   R2
                                        7E  D4 0000B          CLRL    -(SP)
                                        52  DD 0000D          PUSHL   R2
                00000000G  00           03  FB 0000F          CALLS   #3, LIB$SYS_TRNLOG
                         12             50  E8 00016          BLBS    STATUS, 1$              ; 0649
                                        50  DD 00019          PUSHL   STATUS
                                        7E  D4 0001B          CLRL    -(SP)
                           00000000*    8F  DD 0001D          PUSHL   #<<CLI$_FACILITY@16>+4532>
                00000000G  00           03  FB 00023          CALLS   #3, LIB$SIGNAL
                                            04 0002A          RET
                         1B        04   B2  91 0002B 1$:      CMPB    @4(R2), #27             ; 0650
                                        19  12 0002F          BNEQ    2$
                      7E      04    A2      04  C1 00031      ADDL3   #4, 4(R2), -(SP)        ; 0653
                              04    AE      62  3C 00036      MOVZWL  (R2), 4(SP)
                              04    AE      04  C2 0003A      SUBL2   #4, 4(SP)
                                        04  AE  9F 0003E      PUSHAB  4(SP)
                                        52  DD 00041          PUSHL   R2                      ; 0652
                00000000G  00           03  FB 00043          CALLS   #3, STR$COPY_R
                                            04 0004A 2$:      RET                             ; 0654

; Routine Size:  75 bytes,    Routine Base:  $CODE$ + 05C9
```

```
;  488          0655  1 ROUTINE INIT_MBX:       NOVALUE =
;  489          0656  2     BEGIN
;  490          0657  2     MACRO
;  491          0658  2         STR_LOG_1 = 'CHK$CLI_' %,       STR_LOG_2 = '' %,
;  492          0659  2         STR_DEV_1 = '_MBA' %,           STR_DEV_2 = ':' %;
;  493          0660  2     LOCAL
;  494          0661  2         PID:    INITIAL(0);
;  495          0662  2     LOCAL
;  496          0663  2         ITMLST: VECTOR[4] INITIAL(%ALLOCATION(PID), PID, 0, 0);
;  497          0664  2     LOCAL
;  498          0665  2         LOGBUF: $BBLOCK[%CHARCOUNT(STR_LOG_1,STR_LOG_2)+8],
;  499          0666  2 !       DEVBUF: $BBLOCK[%CHARCOUNT(STR_DEV_1,STR_DEV_2)+10],
;  500          0667  2         LOGNAM: $BBLOCK[8] INITIAL(%ALLOCATION(LOGBUF), LOGBUF[BASE_]),
;  501          0668  2 !       DEVNAM: $BBLOCK[8] INITIAL(%ALLOCATION(DEVBUF), DEVBUF[BASE_]),
;  502          0669  2         STATUS;
;  503          0670  2     BIND
;  504          0671  2         ITMLST_W = ITMLST[0]: VECTOR[,WORD];
;  505          0672  2
;  506          0673  2     ! Initialize the message area
;  507          0674  2     !
;  508          0675  2     MSG[MSG_L_ID] = MSG_K_CHKCLI;
;  509          0676  2
;  510          0677  2     ! Get job process information for constructing a logical name.
;  511          0678  2     ! Get the job process id (main process)
;  512          0679  2     ! Get the owner process id (subprocess)
;  513          0680  2     !
;  514          0681  2     ITMLST_W[1] = (IF NOT .CHK_SUBP THEN JPI$_PID ELSE JPI$_OWNER);
;  515        P 0682  2     STATUS = $GETJPIW(
;  516        P 0683  2         EFN     = 0,
;  517          0684  2         ITMLST  = ITMLST[0]);
;  518          0685  2     IF NOT .STATUS THEN RETURN SIGNAL(CLI$_SHR_SYSERROR, 0, .STATUS);
;  519          0686  2
;  520          0687  2     ! Form mailbox logical name
;  521          0688  2     !
;  522        P 0689  2     STATUS = $FAO($DESCRIPTOR(%STRING(STR_LOG_1,'!XL',STR_LOG_2)),
;  523          0690  2             LOGNAM[DSC$W_LENGTH], LOGNAM[BASE_], .PID);
;  524          0691  2     IF NOT .STATUS THEN RETURN SIGNAL(CLI$_SHR_SYSERROR, 0, .STATUS);
;  525          0692  2
;  526          0693  2     ! Create the mailbox (main process)
;  527          0694  2     ! Access the mailbox (subprocess)
;  528          0695  2     !
;  529        P 0696  2     STATUS = $CREMBX(
;  530        P 0697  2         CHAN    = MBX_CHAN,
;  531        P 0698  2         MAXMSG  = %ALLOCATION(MSG),
;  532          0699  2         LOGNAM  = LOGNAM[BASE_]);
;  533          0700  2     IF NOT .STATUS THEN RETURN SIGNAL(CLI$_SHR_SYSERROR, 0, .STATUS);
;  534          0701  2
;  535          0702  2     ! The subprocess doesn't need the unit number
;  536          0703  2     !
;  537          0704  2     IF .CHK_SUBP THEN RETURN SS$_NORMAL;
;  538          0705  2
;  539          0706  2     ! Get the unit number, needed to specify the termination mailbox.
;  540          0707  2     !
;  541          0708  2     ITMLST[0] = DVI$_UNIT ^ 16 + %ALLOCATION(MBX_UNIT);
;  542          0709  2     ITMLST[1] = MBX_UNIT;
;  543        P 0710  2     STATUS = $GETDVIW(
;  544        P 0711  2         EFN             = 0,
```

```
;  545          P 0712  2          CHAN    = .MBX_CHAN,
;  546            0713  2          ITMLST  = ITMLST[0];
;  547            0714  2      IF NOT .STATUS THEN RETURN SIGNAL(CLI$_SHR_SYERROR, 0, .STATUS);
;  548            0715  2
;  549            0716  1      END;


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                00000004  00048 P.AAG:  .LONG   4                                    ;
                                      00000000  00000000  00000000  0004C          .LONG   0, 0, 0                    ;
                                4C  58  21  5F  49  4C  43  24  4B  48  43  00058 P.AAI:  .ASCII  \CHK$CLI_!XL\          ;
                                                                        00063          .BLKB   1
                                                0000000B  00064 P.AAH:  .LONG   11                                   ;
                                                00000000' 00068          .ADDRESS P.AAI                             ;

                                                                .EXTRN  SYS$CREMBX, SYS$GETDVIW

                                                                .PSECT  $CODE$,NOWRT,2

                                                    007C 00000 INIT_MBX:
                                                                .WORD   Save R2,R3,R4,R5,R6                          ; 0655
                                          56      0000'  CF  9E 00002          MOVAB   CHK_SUBP, R6                 ;
                                          5E             28  C2 00007          SUBL2   #40, SP                      ;
                                                         7E  D4 0000A          CLRL    PID                          ; 0656
                                    1C    AE      0000'  CF    10  28 0000C    MOVC3   #16, P.AAG, ITMLST           ; 0663
                                                   20    AE    6E  9E 00013    MOVAB   PID, ITMLST+4                ; 0656
                                                   04    AE    10  D0 00017    MOVL    #16, LOGNAM                  ; 0667
                                                   08    AE      0C  AE  9E 0001B    MOVAB   LOGBUF, LOGNAM+4           ;
                                                   04    A6          04  D0 00020    MOVL    #4, MSG                  ; 0675
                                                   07                66  E8 00024    BLBS    CHK_SUBP, 1$             ; 0681
                                                   50      0319      8F  3C 00027    MOVZWL  #793, R0                 ;
                                                                     05  11 0002C    BRB     2$                       ;
                                                   50      0303      8F  3C 0002E 1$: MOVZWL  #771, R0                 ;
                                                1E    AE                50  B0 00033 2$: MOVW    R0, ITMLST_W+2       ;
                                                                        7E  7C 00037    CLRQ    -(SP)                    ; 0684
                                                                        7E  D4 00039    CLRL    -(SP)                    ;
                                                         28    AE      9F  0003B       PUSHAB  ITMLST               ;
                                                                        7E  7C 0003E    CLRQ    -(SP)                    ;
                                                                        7E  D4 00040    CLRL    -(SP)                    ;
                                          00000000G     00              07  FB 00042    CALLS   #7, SYS$GETJPIW          ;
                                                        52              50  D0 00049    MOVL    R0, STATUS           ;
                                                        63              52  E9 0004C    BLBC    STATUS, 3$           ; 0685
                                                                        6E  DD 0004F    PUSHL   PID                  ; 0690
                                                         08    AE      9F  00051       PUSHAB  LOGNAM               ;
                                                         0C    AE      9F  00054       PUSHAB  LOGNAM               ;
                                                      0000'    CF      9F  00057       PUSHAB  P.AAH                ;
                                          00000000G     00              04  FB 0005B    CALLS   #4, SYS$FAO              ;
                                                        52              50  D0 00062    MOVL    R0, STATUS           ;
                                                        4A              52  E9 00065    BLBC    STATUS, 3$           ; 0691
                                                         04    AE      9F  00068       PUSHAB  LOGNAM               ; 0699
                                                                        7E  7C 0006B    CLRQ    -(SP)                    ;
                                                                        7E  D4 0006D    CLRL    -(SP)                    ;
                                                        7E      0110    8F  3C 0006F    MOVZWL  #272, -(SP)              ;
                                                                 F8     A6  9F 00074    PUSHAB  MBX_CHAN             ;
                                                                        7E  D4 00077    CLRL    -(SP)                    ;
```

```
                00000000G  00              07  FB 00079          CALLS   #7, SYS$CREMBX                                  ;
                           52              50  D0 00080          MOVL    R0, STATUS                                     ;
                           2C              52  E9 00083          BLBC    STATUS, 3$                                     ; 0700
                           3A              66  E8 00086          BLBS    CHK_SUBP, 4$                                   ; 0704
                     1C    AE 000C0004     8F  D0 00089          MOVL    #786436, ITMLST                                ; 0708
                     20    AE        FC    A6  9E 00091          MOVAB   MBX_UNIT, ITMLST+4                             ; 0709
                                           7E  7C 00096          CLRQ    -(SP)                                          ; 0713
                                           7E  7C 00098          CLRQ    -(SP)                                          ;
                                     2C    AE  9F 0009A          PUSHAB  ITMLST                                         ;
                                           7E  D4 0009D          CLRL    -(SP)                                          ;
                           7E        F8    A6  3C 0009F          MOVZWL  MBX_CHAN, -(SP)                                ;
                                           7E  D4 000A3          CLRL    -(SP)                                          ;
                00000000G  00              08  FB 000A5          CALLS   #8, SYS$GETDVIW                                ;
                           52              50  D0 000AC          MOVL    R0, STATUS                                     ;
                           11              52  E8 000AF          BLBS    STATUS, 4$                                     ; 0714
                                           52  DD 000B2 3$:      PUSHL   STATUS                                         ;
                                           7E  D4 000B4          CLRL    -(SP)                                          ;
                                00000000*  8F  DD 000B6          PUSHL   #<<CLI$_FACILITY@16>+4532>                     ;
                00000000G  00              03  FB 000BC          CALLS   #3, LIB$SIGNAL                                 ;
                                              04 000C3 4$:      RET                                                    ; 0716

; Routine Size:  196 bytes,    Routine Base:  $CODE$ + 0614

;  550          0717  1
;  551          0718  1 END
;  552          0719  0 ELUDOM


                                                                .EXTRN  LIB$SIGNAL

                                                                .EXTRN  LIB$SIGNAL

;                                PSECT SUMMARY

;       Name                            Bytes                           Attributes

;  $OWN$                                 285  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                               1752  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                                108  NOVEC,NOWRT,  RD ,NOEXE,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        37         0      1000          00:01.9


; Information:  1
; Warnings:     0
; Errors:       0
```

```
;                                       COMMAND QUALIFIERS

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

; Size:         1752 code + 393 data bytes
; Run Time:       00:30.4
; Elapsed Time:   01:45.8
; Lines/CPU Min:   1417
; Lexemes/CPU-Min: 30690
; Memory Used:  240 pages
; Compilation Complete
```

SUMSLP
MAP

SRTTRNMSG
LIS

DATA
LIS

SUMSHR
MAP

SUMDATA
LIS

UTLCHKCLI
LIS

SUM

SUM
MAP

SUMDEF
MAR

MAIN
LIS

SUMED
LIS

SUMEDIT
LIS

GETCMD
LIS