```
EEEEEEEEEEEEEEE  VVV         VVV  LLL
EEEEEEEEEEEEEEE  VVV         VVV  LLL
EEEEEEEEEEEEEEE  VVV         VVV  LLL
EEE              VVV         VVV  LLL
EEE              VVV         VVV  LLL
EEE              VVV         VVV  LLL
EEE              VVV         VVV  LLL
EEE              VVV         VVV  LLL
EEE              VVV         VVV  LLL
EEEEEEEEEEEE     VVV         VVV  LLL
EEEEEEEEEEEE     VVV         VVV  LLL
EEEEEEEEEEEE     VVV         VVV  LLL
EEE              VVV         VVV  LLL
EEE              VVV         VVV  LLL
EEE              VVV         VVV  LLL
EEE                 VVV   VVV     LLL
EEE                 VVV   VVV     LLL
EEE                 VVV   VVV     LLL
EEEEEEEEEEEEEEE        VVV        LLLLLLLLLLLLLLL
EEEEEEEEEEEEEEE        VVV        LLLLLLLLLLLLLLL
EEEEEEEEEEEEEEE        VVV        LLLLLLLLLLLLLLL
```

```
_$25

Valu
----
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
0200
```

```
WW      WW    000000    RRRRRRRR    KK      KK    QQQQQQ
WW      WW    000000    RRRRRRRR    KK      KK    QQQQQQ
WW      WW  OO      OO  RR      RR  KK      KK  QQ      QQ
WW      WW  OO      OO  RR      RR  KK      KK  QQ      QQ
WW      WW  OO      OO  RR      RR  KK    KK    QQ      QQ
WW      WW  OO      OO  RR      RR  KK    KK    QQ      QQ
WW      WW  OO      OO  RRRRRRRR    KKKKKK      QQ      QQ
WW      WW  OO      OO  RRRRRRRR    KKKKKK      QQ      QQ
WW  WW  WW  OO      OO  RR  RR      KK    KK    QQ  QQ  QQ
WW  WW  WW  OO      OO  RR  RR      KK    KK    QQ  QQ  QQ
WWWW  WWWW  OO      OO  RR    RR    KK      KK  QQ    QQ      ....
WWWW  WWWW  OO      OO  RR    RR    KK      KK  QQ    QQ      ....
WW      WW    000000    RR      RR  KK      KK    QQQQ  QQ    ....
WW      WW    000000    RR      RR  KK      KK    QQQQ  QQ    ....


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
```

: R

```
;     1          0001  0 MODULE workq (IDENT = 'V04-000') =
;     2          0002  1 BEGIN
;     3          0003  1
;     4          0004  1 !
;     5          0005  1 !*****************************************************************************
;     6          0006  1 !*                                                                           *
;     7          0007  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;     8          0008  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;     9          0009  1 !*  ALL RIGHTS RESERVED.                                                      *
;    10          0010  1 !*                                                                           *
;    11          0011  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    12          0012  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    13          0013  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    14          0014  1 !*  COPIES THEREOF MAY NOT BE PROV'DED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    15          0015  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    16          0016  1 !*  TRANSFERRED.                                                              *
;    17          0017  1 !*                                                                           *
;    18          0018  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    19          0019  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    20          0020  1 !*  CORPORATION.                                                              *
;    21          0021  1 !*                                                                           *
;    22          0022  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;    23          0023  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
;    24          0024  1 !*                                                                           *
;    25          0025  1 !*                                                                           *
;    26          0026  1 !***************************************************** *********************
;    27          0027  1
;    28          0028  1 !++
;    29          0029  1 ! FACILITY:  DECnet V2.0 event logger
;    30          0030  1 !
;    31          0031  1 ! ABSTRACT:
;    32          0032  1 !
;    33          0033  1 !       This module contains routines to manage the internal
;    34          0034  1 !       work queue.
;    35          0035  1 !
;    36          0036  1 ! ENVIRONMENT:
;    37          0037  1 !
;    38          0038  1 !       VAX/VMS operating system. unprivileged user mode,
;    39          0039  1 !
;    40          0040  1 ! AUTHOR:  Tim Halvorsen, June 1980
;    41          0041  1 !
;    42          0042  1 ! Modified by:
;    43          0043  1 !
;    44          0044  1 !       V03-001 RPG0001         Bob Grosso              24-Feb-1983
;    45          0045  1 !               Make two minor fixes.  There was one place where a
;    46          0046  1 !               deallocate_vm was done before the REMQUE, and another
;    47          0047  1 !               where there was no deallocation after the REMQUE.
;    48          0048  1 !--
;    49          0049  1
;    50          0050  1 !
;    51          0051  1 ! Include files
;    52          0052  1 !
;    53          0053  1
;    54          0054  1 LIBRARY 'SYS$LIBRARY:STARLET';          ! VAX/VMS common definitions
;    55          0055  1
;    56          0056  1 REQUIRE 'SYS$LIBRARY:UTILDEF';          ! Misc. VMS definitions
;    57          0232  1
```

```
;   58          0233  1 REQUIRE 'LIB$:WQDEF';                 ! Structure definitions
```

```
;    60          0296  1 !
;    61          0297  1 ! Table of contents
;    62          0298  1 !
;    63          0299  1
;    64          0300  1 FORWARD ROUTINE
;    65          0301  1     wkq$add_work_item,                  ! Add a work item
;    66          0302  1     wkq$do_work_item,                   ! Dequeue and execute a work item
;    67          0303  1     wkq$add_timed_work,                 ! Add a timed work item
;    68          0304  1     wkq$cancel_timed_work,              ! Cancel timed work item
;    69          0305  1     timer_ast:          NOVALUE;        ! Timer AST routine
;    70          0306  1
;    71          0307  1 !
;    72          0308  1 ! BUILTIN functions
;    73          0309  1 !
;    74          0310  1
;    75          0311  1 BUILTIN
;    76          0312  1     INSQUE,                             ! INSQUE instruction
;    77          0313  1     REMQUE;                             ! REMQUE instruction
;    78          0314  1
;    79          0315  1 !
;    80          0316  1 ! OWN storage
;    81          0317  1 !
;    82          0318  1
;    83          0319  1 OWN
;    84          0320  1     work_queue:         VECTOR [2]      ! Work queue listhead
;    85          0321  1                         INITIAL(work_queue,work_queue),
;    86          0322  1     timed_queue:        VECTOR [2]      ! Timed work queue listhead
;    87          0323  1                         INITIAL(timed_queue,timed_queue);
;    88          0324  1
;    89          0325  1 !
;    90          0326  1 ! External routines
;    91          0327  1 !
;    92          0328  1
;    93          0329  1 EXTERNAL ROUTINE
;    94          0330  1     lib$get_vm: ADDRESSING_MODE(GENERAL),       ! Allocate storage
;    95          0331  1     lib$free_vm: ADDRESSING_MODE(GENERAL);      ! Deallocate storage
```

```
;    97          0332  1 GLOBAL ROUTINE wkq$add_work_item (action_routine, arg1, arg2) =
;    98          0333  1
;    99          0334  1 !---
;   100          0335  1 !
;   101          0336  1 !       This routine adds a single work item to the work queue.
;   102          0337  1 !       A $WAKE is performed for the first work item added.
;   103          0338  1 !
;   104          0339  1 ! Inputs:
;   105          0340  1 !
;   106          0341  1 !       action_routine = Address of routine to call to perform work.
;   107          0342  1 !       arg1/2 = Arguments to be passed to action routine when called.
;   108          0343  1 !
;   109          0344  1 ! Outputs:
;   110          0345  1 !
;   111          0346  1 !       routine = status code
;   112          0347  1 !---
;   113          0348  1
;   114          0349  2 BEGIN
;   115          0350  2
;   116          0351  2 LOCAL
;   117          0352  2     length,                             ! Length of allocate/deallocation
;   118          0353  2     entry:      REF BBLOCK;             ! Address of work queue entry
;   119          0354  2
;   120          0355  2 length = wqe$c_length;                  ! Length of a work queue entry
;   121          0356  2 return_if_error(LIB$GET_VM(length,entry));       ! Allocate entry
;   122          0357  2 CH$FILL(0,wqe$c_length,.entry);         ! Zero the entry
;   123          0358  2
;   124          0359  2 entry [wqe$l_pc] = .action_routine;     ! Store address of action routine
;   125          0360  2 entry [wqe$l_arg1] = .arg1;             ! then store 2 arguments
;   126          0361  2 entry [wqe$l_arg2] = .arg2;             ! assume both arguments must be given
;   127          0362  2
;   128          0363  2 IF INSQUE(.entry, .work_queue [1])      ! Queue entry at tail of list
;   129          0364  2 THEN
;   130          0365  2     $WAKE();                            ! If first, wakeup process
;   131          0366  2
;   132          0367  2 RETURN true;
;   133          0368  2
;   134          0369  1 END;

                                                .TITLE  WORKQ
                                                .IDENT  \V04-000\

                                                .PSECT  $OWN$,NOEXE,2

                00000000' 00000000' 00000 WORK_QUEUE:
                                                .ADDRESS WORK_QUEUE, WORK_QUEUE                         ;
                00000000' 00000000' 00008 TIMED_QUEUE:
                                                .ADDRESS TIMED_QUEUE, TIMED_QUEUE                       ;

                                                .EXTRN  LIB$GET_VM, LIB$FREE_VM
                                                .EXTRN  SYS$WAKE

                                                .PSECT  $CODE$,NOWRT,2

                                  007C 00000    .ENTRY  WKQ$ADD_WORK_ITEM, Save R2,R3,R4,R5,R6          ; 0332
                5E                08  C2 00002 SUBL2   #8, SP                                          ;
```

```
                        04   AE          18  D0 00005          MOVL    #24, LENGTH                                    ; 0355
                                         5E  DD 00009          PUSHL   SP                                             ; 0356
                                     08  AE  9F 0000B          PUSHAB  LENGTH
                 00000000G   00          02  FB 0000E          CALLS   #2, LIB$GET_VM
                             26          50  E9 00015          BLBC    STATUS, 2$
                             56          6E  D0 00018          MOVL    ENTRY, R6                                      ; 0357
          18            00   6E          00  2C 0001B          MOVC5   #0, (SP), #0, #24, (R6)
                                           66     00020
                        08   A6      04  AC  7D 00021          MOVQ    ACTION_ROUTINE, 8(R6)                          ; 0359
                        10   A6      0C  AC  D0 00026          MOVL    ARG2, T6(R6)                                   ; 0361
                      0000'  DF          66  0E 0002B          INSQUE  (R6), @WORK_QUEUE+4                            ; 0363
                                         09  12 00030          BNEQ    1$
                                         7E  7C 00032          CLRQ    -(SP)                                          ; 0365
                 00000000G   00          02  FB 00034          CALLS   #2, SYS$WAKE
                             50          01  D0 0003B 1$:      MOVL    #1, R0                                         ; 0367
                                             04 0003E 2$:      RET                                                    ; 0369

; Routine Size:  63 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  136          0370  1 GLOBAL ROUTINE wkq$do_work_item =
;  137          0371  1
;  138          0372  1 !---
;  139          0373  1 !
;  140          0374  1 !       This routine dequeues the next work item to be performed
;  141          0375  1 !       and calls the action routine associated with the item.
;  142          0376  1 !
;  143          0377  1 ! Inputs:
;  144          0378  1 !
;  145          0379  1 !       None
;  146          0380  1 !
;  147          0381  1 ! Outputs:
;  148          0382  1 !
;  149          0383  1 !       routine = True if successful, false if nothing in queue
;  150          0384  1 !---
;  151          0385  1
;  152          0386  2 BEGIN
;  153          0387  2
;  154          0388  2 LOCAL
;  155          0389  2     length,                         ! Length of work queue entry
;  156          0390  2     entry:      REF BBLOCK;         ! Address of entry in queue
;  157          0391  2
;  158          0392  2 IF REMQUE(.work_queue [0], entry)    ! Remove first entry from queue
;  159          0393  2 THEN
;  160          0394  2     RETURN false;                   ! If none, return unsuccessful
;  161          0395  2
;  162          0396  2 (.entry [wqe$l_pc])(.entry [wqe$l_arg1], .entry [wqe$l_arg2]); ! Call routine
;  163          0397  2
;  164          0398  2 length = wqe$c_length;
;  165          0399  2 LIB$FREE_VM(length, entry);          ! Deallocate entry
;  166          0400  2
;  167          0401  2 RETURN true;
;  168          0402  2
;  169          0403  1 END;


                                        0000 00000          .ENTRY  WKQ$DO_WORK_ITEM, Save nothing      ; 0370
                                5E      04   C2 00002          SUBL2   #4, SP
                                7E  0000' DF   0F 00005          REMQUE  @WORK_QUEUE, ENTRY                  ; 0392
                                            1F   1D 0000A          BVS     1$
                                50          6E   D0 0000C          MOVL    ENTRY, R0                       ; 0396
                                7E      0C  A0   7D 0000F          MOVQ    12(R0), -(SP)
                            08  B0          02   FB 00013          CALLS   #2, @8(R0)
                            04  AE          18   D0 00017          MOVL    #24, LENGTH                     ; 0398
                                5E      DD 0001B                   PUSHL   SP                              ; 0399
                                    08  AE  9F   0001D          PUSHAB  LENGTH
                    00000000G   00          02   FB 00020          CALLS   #2, LIB$FREE_VM
                                50          01   D0 00027          MOVL    #1, R0                          ; 0401
                                            04   0002A          RET
                                            50   D4 0002B 1$:     CLRL    R0                              ; 0403
                                            04   0002D          RET

; Routine Size:  46 bytes,    Routine Base:  $CODE$ + 003F
```

LPM

%SE

OWN

PSE

! T
! a
!

```
;  171     0404  1 GLOBAL ROUTINE wkq$add_timed_work (action_routine, arg1, arg2, time, reqid) =
;  172     0405  1
;  173     0406  1 !---
;  174     0407  1 !
;  175     0408  1 !       This routine adds a timed work item to the work queue
;  176     0409  1 !       to be executed at the specified time.  Any outstanding
;  177     0410  1 !       work item for the same reqid is canceled before the new
;  178     0411  1 !       one is added.
;  179     0412  1 !
;  180     0413  1 ! Inputs:
;  181     0414  1 !
;  182     0415  1 !       action_routine = Address of routine to call to perform work.
;  183     0416  1 !       arg1/2 = Arguments to be passed to action routine when called.
;  184     0417  1 !       time = Delta or absolute time at which the work item should be done.
;  185     0418  1 !       reqid = Request identification to be associated with the time based
;  186     0419  1 !               work item.  There may be no more than one work item in the
;  187     0420  1 !               work queue with this request id.
;  188     0421  1 !
;  189     0422  1 ! Outputs:
;  190     0423  1 !
;  191     0424  1 !       routine = status code
;  192     0425  1 !---
;  193     0426  1
;  194     0427  2 BEGIN
;  195     0428  2
;  196     0429  2 LOCAL
;  197     0430  2     status,
;  198     0431  2     length,                         ! Length of allocate/deallocation
;  199     0432  2     entry:      REF BBLOCK,         ! Address of work queue entry
;  200     0433  2     p:          REF BBLOCK;         ! Address of entry being scanned
;  201     0434  2
;  202     0435  2 length = wqe$c_length;              ! Length of a work queue entry
;  203     0436  2 return_if_error(LIB$GET_VM(length,entry));      ! Allocate entry
;  204     0437  2 CH$FILL(0,wqe$c_length,.entry);     ! Zero the entry
;  205     0438  2
;  206     0439  2 entry [wqe$l_pc] = .action_routine; ! Store address of action routine
;  207     0440  2 entry [wqe$l_arg1] = .arg1;         ! then store 2 arguments
;  208     0441  2 entry [wqe$l_arg2] = .arg2;         ! assume both arguments must be given
;  209     0442  2 entry [wqe$l_reqid] = .reqid;       ! Store request id
;  210     0443  2
;  211     0444  2 wkq$cancel_timed_work(.reqid);      ! Cancel outstanding request (if any)
;  212     0445  2
;  213     0446  2 INSQUE(.entry, .timed_queue [1]);   ! Insert at end of queue
;  214     0447  2
;  215     0448  2 status = $SETIMR(DAYTIM=.time, ASTADR=timer_ast, REQIDT=.reqid);
;  216     0449  2
;  217     0450  2 IF NOT .status                      ! If error occurred setting timer,
;  218     0451  2 THEN
;  219     0452  3     BEGIN
;  220     0453  3     REMQUE(.entry, entry);          ! Remove entry from queue
;  221     0454  3     LIB$FREE_VM(length,entry);      ! Deallocate the storage
;  222     0455  3     RETURN .status;                 ! and return with status
;  223     0456  2     END;
;  224     0457  2
;  225     0458  2 RETURN true;
;  226     0459  2
;  227     0460  1 END;
```

```
                                                        .EXTRN  SYS$SETIMR

                                                 007C 00000     .ENTRY  WKQ$ADD_TIMED_WORK, Save R2,R3,R4,R5,R6      ; 0404
                                5E               08  C2 00002  SUBL2   #8, SP                                       ;
                             04 AE               18  D0 00005  MOVL    #24, LENGTH                                  ; 0435
                                                 5E  DD 00009  PUSHL   SP                                           ; 0436
                                              08 AE  9F 0000B  PUSHAB  LENGTH                                       ;
                    00000000G   00               02  FB 0000E  CALLS   #2, LIB$GET_VM                              ;
                                54               50  E9 00015  BLBC    STATUS, 2$                                   ;
                                56               6E  D0 00018  MOVL    ENTRY, R6                                    ; 0437
                18          00  6E               00  2C 0001B  MOVC5   #0, (SP), #0, #24, (R6)                      ;
                                                 66     00020                                                       ;
                             08 A6           04 AC  7D 00021  MOVQ    ACTION_ROUTINE, 8(R6)                        ; 0439
                             10 A6           0C AC  D0 00026  MOVL    ARG2, 16(R6)                                 ; 0441
                             14 A6           14 AC  D0 0002B  MOVL    REQID, 20(R6)                                ; 0442
                                             14 AC  DD 00030  PUSHL   REQID                                        ; 0444
                          0000V CF                  01  FB 00033  CALLS   #1, WKQ$CANCEL_TIMED_WORK                ;
                          0000' DF                  66  0E 00038  INSQUE  (R6), @TIMED_QUEUE+4                     ; 0446
                                             14 AC  DD 0003D  PUSHL   REQID                                        ; 0448
                                           0000V CF  9F 00040  PUSHAB  TIMER_AST                                    ;
                                             10 AC  DD 00044  PUSHL   TIME                                         ;
                                                 7E  D4 00047  CLRL    -(SP)                                        ;
                    00000000G   00               04  FB 00049  CALLS   #4, SYS$SETIMR                             ;
                                52               50  D0 00050  MOVL    R0, STATUS                                   ;
                                13               52  E8 00053  BLBS    STATUS, 1$                                   ; 0450
                                7E               9E  0F 00056  REMQUE  @ENTRY, ENTRY                                ; 0453
                                                 5E  DD 00059  PUSHL   SP                                           ; 0454
                                              08 AE  9F 0005B  PUSHAB  LENGTH                                       ;
                    00000000G   00               02  FB 0005E  CALLS   #2, LIB$FREE_VM                             ;
                                50               52  D0 00065  MOVL    STATUS, R0                                   ; 0455
                                                 04 00068  RET                                                      ;
                                50               01  D0 00069  1$:    MOVL    #1, R0                               ; 0458
                                                 04 0006C  2$:    RET                                              ; 0460

; Routine Size:  109 bytes,    Routine Base:  $CODE$ + 006D
```

```
;  229          0461  1 GLOBAL ROUTINE wkq$cancel_timed_work (reqid) =
;  230          0462  1
;  231          0463  1 !---
;  232          0464  1 !
;  233          0465  1 !       This routine cancels a timed work item in the work queue
;  234          0466  1 !       from being executed at the specified time.
;  235          0467  1 !
;  236          0468  1 ! Inputs:
;  237          0469  1 !
;  238          0470  1 !       reqid = Request identification associated with the work item.
;  239          0471  1 !
;  240          0472  1 ! Outputs:
;  241          0473  1 !
;  242          0474  1 !       routine = True if request dequeued, false if not found in work queue.
;  243          0475  1 !---
;  244          0476  1
;  245          0477  2 BEGIN
;  246          0478  2
;  247          0479  2 LOCAL
;  248          0480  2     length,
;  249          0481  2     p:          REF BBLOCK;             ! Address of entry being scanned
;  250          0482  2
;  251          0483  2 length = wqe$c_length;                  ! Length of a work queue entry
;  252          0484  2 p = .timed_queue [0];                   ! Get at first entry in timed queue
;  253          0485  2
;  254          0486  2 WHILE .p NEQ timed_queue [0]            ! Until the end of list
;  255          0487  2 DO
;  256          0488  3     BEGIN
;  257          0489  3     IF .p [wqe$l_reqid] EQL .reqid      ! If already a request with this reqid,
;  258          0490  3     THEN
;  259          0491  4         BEGIN
;  260          0492  4         $CANTIM(REQIDT=.reqid);         ! Cancel the previous request
;  261          0493  4         REMQUE(.p, p);                  ! Remove entry from queue
;  262          0494  4         LIB$FREE_VM(length,p);          ! Deallocate the storage
;  263          0495  4         RETURN true;                    ! Return successful
;  264          0496  3         END;
;  265          0497  3     p = .p [wqe$l_flink];               ! Skip to next in chain
;  266          0498  2     END;
;  267          0499  2
;  268          0500  2 RETURN false;                           ! Indicate request not in work queue
;  269          0501  2
;  270          0502  1 END;


                                                        .EXTRN  SYS$CANTIM

                                        0004 00000      .ENTRY  WKQ$CANCEL_TIMED_WORK, Save R2          ; 0461
                                     18 DD 00002        PUSHL   #24                                     ; 0483
                                0000' CF DD 00004       PUSHL   TIMED_QUEUE                             ; 0484
                             52     6E D0 00008 1$:     MOVL    P, R2                                   ; 0486
                             50 0000' CF 9E 0000B       MOVAB   TIMED_QUEUE, R0                         ;
                             50     52 D1 00010         CMPL    R2, R0                                  ;
                                    2A 13 00013         BEQL    3$                                      ;
                          04 AC   14    A2 D1 00015     CMPL    20(R2), REQID                           ; 0489
                                        1F 12 0001A     BNEQ    2$                                      ;
                                        7E D4 0001C     CLRL    -(SP)                                   ; 0492
```

```
                          04 AC  DD 0001E          PUSHL   REQID                                          ;
                00000000G 00 02  FB 0C021          CALLS   #2, SYS$CANTIM                               ;
                          6E 62  0F 00028          REMQUE  (R2), P                                      ; 0493
                             5E  DD 0002B          PUSHL   SP                                           ; 0494
                          08 AE  9F 0002D          PUSHAB  LENGTH                                       ;
                00000000G 00 02  FB 00030          CALLS   #2, LIB$FREE_VM                              ;
                          50 01  D0 00037          MOVL    #1, R0                                       ; 0495
                             04  04 0003A          RET                                                  ;
                             9E  DD 0003B 2$:     PUSHL   @P                                           ; 0497
                             C9  11 0003D         BRB     1$                                           ; 0486
                             50  D4 0003F 3$:     CLRL    R0                                           ; 0500
                                 04 00041         RET                                                  ; 0502

; Routine Size:  66 bytes,    Routine Base:  $CODE$ + 00DA
```

```
;   272          0503  1 ROUTINE timer_ast (reqid): NOVALUE =
;   273          0504  1
;   274          0505  1 !---
;   275          0506  1 !
;   276          0507  1 !       This AST is called when a timer has run out for a work
;   277          0508  1 !       queue item.  The action routine associated with the
;   278          0509  1 !       item is called.
;   279          0510  1 !
;   280          0511  1 ! Inputs:
;   281          0512  1 !
;   282          0513  1 !       reqid = Request identification for this timer request
;   283          0514  1 !
;   284          0515  1 ! Outputs:
;   285          0516  1 !
;   286          0517  1 !       None
;   287          0518  1 !---
;   288          0519  1
;   289          0520  2 BEGIN
;   290          0521  2
;   291          0522  2 LOCAL
;   292          0523  2     entry:      REF BBLOCK,             ! Address of work entry begin scanned
;   293          0524  2     length;                             ! Length of work queue entry
;   294          0525  2
;   295          0526  2 entry = .timed_queue [0];               ! Get first timed entry
;   296          0527  2
;   297          0528  2 WHILE .entry NEQ timed_queue [0]        ! Until end of queue
;   298          0529  2 DO
;   299          0530  3     BEGIN
;   300          0531  3     IF .reqid EQL .entry [wqe$l_reqid]  ! If matching item is found,
;   301          0532  3     THEN
;   302          0533  4         BEGIN
;   303          0534  4         REMQUE(.entry, entry);          ! Remove entry from queue
;   304          0535  4
;   305          0536  4         wkq$add_work_item(.entry [wqe$l_pc],    ! Insert onto normal work queue
;   306          0537  4                 .entry [wqe$l_arg1],
;   307          0538  4                 .entry [wqe$l_arg2]);
;   308          0539  4
;   309          0540  4         length = wqe$c_length;
;   310          0541  4         LIB$FREE_VM(length, entry);     ! Deallocate entry
;   311          0542  4         RETURN;
;   312          0543  3         END;
;   313          0544  3     entry = .entry [wqe$l_flink];       ! Skip to next entry in chain
;   314          0545  2     END;
;   315          0546  2
;   316          0547  2 !
;   317          0548  2 !       A timer has run out without a corresponding work item in the queue.
;   318          0549  2 !       Ignore the timer AST.
;   319          0550  2 !
;   320          0551  2
;   321          0552  2 RETURN;
;   322          0553  2
;   323          0554  1 END;
```

```
                                0000 00000 TIMER_AST:
                                                .WORD   Save nothing                                    ; 0503
                          5E                    04  C2 00002          SUBL2   #4, SP                  ;
                                0000'           CF  DD 00005          PUSHL   TIMED_QUEUE             ; 0526
                          51                    6E  D0 00009 1$:      MOVL    ENTRY, R1               ; 0528
                          50    0000'           CF  9E 0000C          MOVAB   TIMED_QUEUE, R0         ;
                          50                    51  D1 00011          CMPL    R1, R0                  ;
                                                2E  13 00014          BEQL    3$                      ;
                       14 A1          04        AC  D1 00016          CMPL    REQID, 20(R1)           ; 0531
                                                23  12 0001B          BNEQ    2$                      ;
                          6E                    61  0F 0001D          REMQUE  (R1), ENTRY             ; 0534
                          50                    6E  D0 00020          MOVL    ENTRY, R0               ; 0538
                          7E            0C      A0  7D 00023          MOVQ    12(R0), -(SP)           ; 0537
                                        08      A0  DD 00027          PUSHL   8(R0)                   ; 0536
                    FEB5  CF                    03  FB 0002A          CALLS   #3, WKQ$ADD_WORK_ITEM   ;
                      04  AE                    18  D0 0002F          MOVL    #24, LENGTH             ; 0540
                                                5E  DD 00033          PUSHL   SP                      ; 0541
                                        08      AE  9F 00035          PUSHAB  LENGTH                  ;
                00000000G 00                    02  FB 00038          CALLS   #2, LIB$FREE_VM         ;
                                                04  00003F           RET                             ; 0533
                                                9E  DD 00040 2$:      PUSHL   @ENTRY                  ; 0544
                                                C5  11 00042          BRB     1$                      ; 0528
                                                04  00044 3$:       RET                             ; 0554

; Routine Size:  69 bytes,    Routine Base:  $CODE$   011C
```

```
;  325          0555  1 END
;  326          0556  0 ELUDOM


;                                  PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  $OWN$                           16  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                         353  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]STARLET.L32;1      9776        10         0       581          00:01.0


;                                       COMMAND QUALIFIERS

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

; Size:          353 code + 16 data bytes
; Run Time:          00:08.8
; Elapsed Time:      00:17.8
; Lines/CPU Min:      3773
; Lexemes/CPU-Min: 16561
; Memory Used:  67 pages
; Compilation Complete
```

LPMULT
B32

DRMAST
MAR

ADDRIVER
MAR

TDRIVER
MAR

USSTEST
MAR

GB_SECUFO
MAR

USSDISP
MAR

DOD_ERAPAT
MAR

LBRMAC
MAR

XADRIVER
MAR

LABIOCIN
MAR

DRSLV
MAR

DTE_DF03
MAR

WORKQ
LIS

SCRFT
MAR

EXAMPLES