```
   CCCCCCCCCCC  LLL              IIIIIIIII     UUU         UUU  TTTTTTTTTTTTTTT  LLL
   CCCCCCCCCCC  LLL              IIIIIIIII     UUU         UUU  TTTTTTTTTTTTTTT  LLL
   CCCCCCCCCCC  LLL              IIIIIIIII     UUU         UUU  TTTTTTTTTTTTTTT  LLL
CCC             LLL                 III      UUU         UUU        TTT        LLL
CCC             LLL                 III      UUU         UUU        TTT        LLL
CCC             LLL                 III      UUU         UUU        TTT        LLL
CCC             LLL                 III      UUU         UUU        TTT        LLL
CCC             LLL                 III      UUU         UUU        TTT        LLL
CCC             LLL                 III      UUU         UUU        TTT        LLL
CCC             LLL                 III      UUU         UUU        TTT        LLL
CCC             LLL                 III      UUU         UUU        TTT        LLL
CCC             LLL                 III      UUU         UUU        TTT        LLL
CCC             LLL                 III      UUU         UUU        TTT        LLL
CCC             LLL                 III      UUU         UUU        TTT        LLL
CCC             LLL                 III      UUU         UUU        TTT        LLL
CCC             LLL                 III      UUU         UUU        TTT        LLL
   CCCCCCCCCCC  LLLLLLLLLLLLLLL     IIIIIIIII     UUUUUUUUUUUUUUU        TTT        LLLLLLLLLLLLLLL
   CCCCCCCCCCC  LLLLLLLLLLLLLLL     IIIIIIIII     UUUUUUUUUUUUUUU        TTT        LLLLLLLLLLLLLLL
   CCCCCCCCCCC  LLLLLLLLLLLLLLL     IIIIIIIII     UUUUUUUUUUUUUUU        TTT        LLLLLLLLLLLLLLL
```

```
  SSSSSSSS  EEEEEEEEEE  TTTTTTTTTT    AAAAAA     CCCCCCCC  NN      NN  TTTTTTTTTT  NN      NN    GGGGGGGG
  SSSSSSSS  EEEEEEEEEE  TTTTTTTTTT    AAAAAA     CCCCCCCC  NN      NN  TTTTTTTTTT  NN      NN    GGGGGGGG
SS          EE              TT      AA    AA   CC        NN      NN      TT      NN      NN  GG
SS          EE              TT      AA    AA   CC        NN      NN      TT      NN      NN  GG
SS          EE              TT      AA    AA   CC        NNNN    NN      TT      NNNN    NN  GG
SS          EE              TT      AA    AA   CC        NNNN    NN      TT      NNNN    NN  GG
  SSSSSS    EEEEEEEE        TT      AA    AA   CC        NN  NN  NN      TT      NN  NN  NN  GG
  SSSSSS    EEEEEEEE        TT      AA    AA   CC        NN  NN  NN      TT      NN  NN  NN  GG
        SS  EE              TT      AAAAAAAAAA CC        NN    NNNN      TT      NN    NNNN  GG  GGGGGG
        SS  EE              TT      AAAAAAAAAA CC        NN    NNNN      TT      NN    NNNN  GG  GGGGGG
        SS  EE              TT      AA    AA   CC        NN      NN      TT      NN      NN  GG      GG      ....
        SS  EE              TT      AA    AA   CC        NN      NN      TT      NN      NN  GG      GG      ....
SSSSSSSS    EEEEEEEEEE      TT      AA    AA     CCCCCCCC  NN      NN      TT      NN      NN    GGGGGG      ....
SSSSSSSS    EEEEEEEEEE      TT      AA    AA     CCCCCCCC  NN      NN      TT      NN      NN    GGGGGG      ....

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 setacntng ( IDENT = 'V04-000',
;     2          0002  0                  ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL=LONG_RELATIVE)
;     3          0003  0                  ) =
;     4          0004  1 BEGIN
;     5          0005  1
;     6          0006  1 !
;     7          0007  1 !*****************************************************************************
;     8          0008  1 !*                                                                           *
;     9          0009  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;    10          0010  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;    11          0011  1 !*  ALL RIGHTS RESERVED.                                                      *
;    12          0012  1 !*                                                                           *
;    13          0013  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
;    14          0014  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
;    15          0015  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
;    16          0016  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
;    17          0017  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
;    18          0018  1 !*  TRANSFERRED.                                                              *
;    19          0019  1 !*                                                                           *
;    20          0020  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
;    21          0021  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT     *
;    22          0022  1 !*  CORPORATION.                                                              *
;    23          0023  1 !*                                                                           *
;    24          0024  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
;    25          0025  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
;    26          0026  1 !*                                                                           *
;    27          0027  1 !*                                                                           *
;    28          0028  1 !*****************************************************************************
;    29          0029  1
;    30          0030  1 !++
;    31          0031  1 ! FACILITY:  SET Command
;    32          0032  1 !
;    33          0033  1 ! ABSTRACT:
;    34          0034  1 !
;    35          0035  1 !       This module implements the DCL command SET ACCOUNTING.
;    36          0036  1 !
;    37          0037  1 ! ENVIRONMENT:
;    38          0038  1 !
;    39          0039  1 !       VAX/VMS operating system, user mode
;    40          0040  1 !
;    41          0041  1 ! AUTHOR:  Gerry Smith                              15-Mar-1983
;    42          0042  1 !
;    43          0043  1 ! Modified by:
;    44          0044  1 !
;    45          0045  1 !       V03-003 DAS0001         David Solomon           09-Jul-1984
;    46          0046  1 !               Fix truncation errors; make nonexternal refs LONG_RELATIVE.
;    47          0047  1 !
;    48          0048  1 !       V03-002 GAS0156         Gerry Smith             24-Jul-1983
;    49          0049  1 !               Fix error signaling for SET ACCOUNT/NEW.
;    50          0050  1 !
;    51          0051  1 !       V03-001 GAS0144         Gerry Smith             22-Jun-1983
;    52          0052  1 !               Convert to new SNDJBC service.
;    53          0053  1 !
;    54          0054  1 !--
```

```
;    56          0055  1 !
;    57          0056  1 ! Include files
;    58          0057  1 !
;    59          0058  1 LIBRARY 'SYS$LIBRARY:STARLET';                  ! VAX/VMS common definitions
;    60          0059  1
```

```
;    62          0060  1 !
;    63          0061  1 ! Declare some storage for tables
;    64          0062  1 !
;    65          0063  1 OWN
;    66          0064  1     option_string : VECTOR[2]                       ! ASCII storage for
;    67          0065  1                     INITIAL(%ASCID 'ENABLE',        ! qualifiers
;    68          0066  1                             %ASCID 'DISABLE'),
;    69          0067  1
;    70          0068  1
;    71          0069  1     option_code : VECTOR[2]                         ! Corresponding codes
;    72          0070  1                   INITIAL(sjc$_start_accounting,
;    73          0071  1                           sjc$_stop_accounting),
;    74          0072  1
;    75          0073  1 !
;    76          0074  1 ! List the accounting types, in ASCII, and the corresponding bitmasks
;    77          0075  1 !
;    78          0076  1 ! The following tables contain data in a specific order, and that order
;    79          0077  1 ! is the $BITPOSITION order of the ACM$V_ accounting types fields.  If new
;    80          0078  1 ! accounting types are added to LIB:ACMDEF and SJCDEF, then corresponding
;    81          0079  1 ! entriesshould be added to the tables here.
;    82          0080  1 !               *******Note that currently, the ACM$V_  fields
;    83          0081  1 !               start at bit position 0 and are incremented by 1
;    84          0082  1 !               for each accounting  type.  If this changes in the
;    85          0083  1 !               future, it may be necessary to change the way this
;    86          0084  1 !               module, specifically the logging part, is
;    87          0085  1 !               implemented.
;    88          0086  1 !
;    89          0087  1 !
;    90          0088  1     acc_name : VECTOR[10]                                   ! ASCII descriptors for
;    91          0089  1                INITIAL(%ASCID 'PROCESS,',                   ! accounting categories
;    92          0090  1                        %ASCID 'IMAGE,',
;    93          0091  1                        %ASCID 'INTERACTIVE,',
;    94          0092  1                        %ASCID 'LOGIN_FAILURE,',
;    95          0093  1                        %ASCID 'SUBPROCESS,',
;    96          0094  1                        %ASCID 'DETACHED,',
;    97          0095  1                        %ASCID 'BATCH,',
;    98          0096  1                        %ASCID 'NETWORK,',
;    99          0097  1                        %ASCID 'PRINT,',
;   100          0098  1                        %ASCID 'MESSAGE,'),
;   101          0099  1
;   102          0100  1     acc_code : VECTOR[10]                                   ! Corresponding masks
;   103          0101  1                INITIAL(sjc$m_acct_process,
;   104          0102  1                        sjc$m_acct_image,
;   105          0103  1                        sjc$m_acct_interactive,
;   106          0104  1                        sjc$m_acct_login_failure,
;   107          0105  1                        sjc$m_acct_subprocess,
;   108          0106  1                        sjc$m_acct_detached,
;   109          0107  1                        sjc$m_acct_batch,
;   110          0108  1                        sjc$m_acct_network,
;   111          0109  1                        sjc$m_acct_print,
;   112          0110  1                        sjc$m_acct_message);
;   113          0111  1
;   114          0112  1
```

```
;  116          0113  1 !
;  117          0114  1 ! Table of contents
;  118          0115  1 !
;  119          0116  1
;  120          0117  1 FORWARD ROUTINE
;  121          0118  1     set$accounting: NOVALUE,            ! Main module of SET ACCOUNTING
;  122          0119  1     process_request : NOVALUE,          ! Process qualifiers
;  123          0120  1     send_request;                       ! Send request to acc. mgr.
;  124          0121  1
;  125          0122  1 !
;  126          0123  1 ! Declare the accounting manager flags in the exec
;  127          0124  1 !
;  128          0125  1 EXTERNAL
;  129          0126  1     exe$gl_acmflags;
;  130          0127  1
;  131          0128  1 !
;  132          0129  1 ! External routines
;  133          0130  1 !
;  134          0131  1 EXTERNAL ROUTINE
;  135          0132  1     str$append,                         ! Append one string to another
;  136          0133  1     cli$get_value,                      ! Get value from CLI
;  137          0134  1     cli$present;                        ! See if qualifier is present
;  138          0135  1
;  139          0136  1 !
;  140          0137  1 ! Declare literals defined elsewhere
;  141          0138  1 !
;  142          0139  1 EXTERNAL LITERAL
;  143          0140  1     set$_accenab,                       ! List currently enabled types
;  144          0141  1     set$_accdisab,                      ! Accounting now disabled
;  145          0142  1     set$_writeerr,                      ! Error modifying accntng. params
;  146          0143  1     set$_newfile,                       ! New accounting file created
;  147          0144  1     set$_nonewfile;                     ! New acc. file not created
;  148          0145  1
```

```
;  150          0146  1 GLOBAL ROUTINE set$accounting : NOVALUE =
;  151          0147  2 BEGIN
;  152          0148  2
;  153          0149  2 !++
;  154          0150  2 ! Functional description
;  155          0151  2 !
;  156          0152  2 !       This is the routine for the SET ACCOUNTING command.  It is called
;  157          0153  2 !       from the SET command processor, and enables/disables certain types
;  158          0154  2 !       of accounting, as well as starting a new accounting file.
;  159          0155  2 !
;  160          0156  2 ! Inputs
;  161          0157  2 !       None
;  162          0158  2 !
;  163          0159  2 ! Outputs
;  164          0160  2 !       None
;  165          0161  2 !
;  166          0162  2 !----
;  167          0163  2
;  168          0164  2 LOCAL
;  169          0165  2     status,                                 ! Status return
;  170          0166  2     log : BYTE,                             ! Tell whether to log or not
;  171          0167  2     flags : VOLATILE,                       ! Flags to show what was done
;  172          0168  2     buffer : VECTOR[4];                     ! Message buffer to send request
;  173          0169  2
;  174          0170  2 !
;  175          0171  2 ! See if logging is required.
;  176          0172  2 !
;  177          0173  2 log = cli$present(%ASCID 'LOG');
;  178          0174  2
;  179          0175  2 !
;  180          0176  2 ! If something is to be enabled, process the qualifiers, then request
;  181          0177  2 ! the change from the accounting manager.  If something went wrong,
;  182          0178  2 ! signal an error.  Otherwise, if the operation is to be logged, issue
;  183          0179  2 ! an informational message.
;  184          0180  2 !
;  185          0181  2 INCR i FROM 0 TO 1 DO
;  186          0182  3     BEGIN
;  187          0183  3     IF cli$present(.option_string[.i])
;  188          0184  3     THEN
;  189          0185  4         BEGIN
;  190          0186  4 !
;  191          0187  4 ! Call the routine to determine what is to be enabled/disabled.
;  192          0188  4 !
;  193          0189  4         flags = 0;                                      ! Clear all flags
;  194          0190  4         process_request(.option_string[.i], flags);     ! See what to change
;  195          0191  4
;  196          0192  4 !
;  197          0193  4 ! If nothing to set, then put a zero at the end of the itemlist.
;  198          0194  4 !
;  199          0195  4         IF .flags EQL 0                                 ! If no flags set,
;  200          0196  4         THEN buffer[0] = 0                              ! then all done
;  201          0197  4
;  202          0198  4 !
;  203          0199  4 ! If there is something to enable/disable, then add another item to
;  204          0200  4 ! the itemlist, the accounting types.
;  205          0201  4 !
;  206          0202  4         ELSE
```

```
;  207     0203  5            BEGIN
;  208     0204  5            buffer[0] = sjc$_accounting_types^16         ! Accounting types
;  209     0205  5                        OR 4;                           ! are a longword
;  210     0206  5            buffer[1] = flags;                          ! located here
;  211     0207  5            buffer[2] = 0;                              ! no return length
;  212     0208  5            buffer[3] = 0;                              ! end of list
;  213     0209  4            END;
;  214     0210  4
;  215     0211  4 !
;  216     0212  4 ! Call the routine to actually send the request to the job controller.
;  217     0213  4 !
;  218     0214  5            IF NOT (status = send_request(.option_code[.i], ! This is the function,
;  219     0215  5                                           buffer))         ! this is the itemlist
;  220     0216  4            THEN
;  221     0217  5                BEGIN                                   ! If an error,
;  222     0218  5                SIGNAL(set$_writeerr, 1,                ! tell user
;  223     0219  5                       %ASCID 'accounting parameters',
;  224     0220  5                       .status);
;  225     0221  5                RETURN;                                 ! and go away
;  226     0222  4                END;
;  227     0223  3            END;
;  228     0224  2        END;
;  229     0225  2
;  230     0226  2 !
;  231     0227  2 ! If /LOG was specified, then display the current accounting types enabled.
;  232     0228  2 !
;  233     0229  2 IF .log
;  234     0230  2 THEN
;  235     0231  3     BEGIN
;  236     0232  3     LOCAL
;  237     0233  3         types : $BBLOCK[dsc$c_s_bln],   ! Place to build the types string
;  238     0234  3         acmflags : BITVECTOR[32];       ! Temporary place for accntng. flags
;  239     0235  3     $init_dyndesc(types);               ! Make a dynamic descriptor
;  240     0236  3     acmflags = .exe$gl_acmflags;        ! Get a copy of current settings.
;  241     0237  3     INCR i FROM 0 TO 9 DO
;  242     0238  3         IF .acmflags[.i]
;  243     0239  3         THEN str$append(types, .acc_name[.i]);
;  244     0240  3     IF .types[dsc$w_length] EQL 0       ! If nothing set, all disabled
;  245     0241  3     THEN SIGNAL(set$_accdisab)
;  246     0242  3     ELSE                                ! Otherwise, strip trailing ","
;  247     0243  4         BEGIN                           ! and display enabled types
;  248     0244  4         types[dsc$w_length] = .types[dsc$w_length] - 1;
;  249     0245  4         SIGNAL(set$_accenab, 1, types);
;  250     0246  3         END;
;  251     0247  2     END;
;  252     0248  2
;  253     0249  2 !
;  254     0250  2 ! If a new accounting file is requested, try to do that.  If something went
;  255     0251  2 ! wrong, signal an error.  Otherwise, if the operation is to be logged,
;  256     0252  2 ! issue an informational message.
;  257     0253  2 !
;  258     0254  2 IF cli$present(%ASCID 'NEW_FILE')
;  259     0255  2 THEN
;  260     0256  3     BEGIN
;  261     0257  3     buffer[0] = sjc$_new_version^16;            ! Open a new file
;  262     0258  3     buffer[1] = 0;
;  263     0259  3     buffer[2] = 0;
```

```
;  264        0260  3       buffer[3] = 0;
;  265        0261  4       IF NOT (status = send_request(.option_code[0],
;  266        0262  4                                     buffer))
;  267        0263  3       THEN SIGNAL(set$_nonewfile, 0,                        ! If an error,
;  268        0264  3                   .status)                                  ! tell user
;  269        0265  3       ELSE IF .log                                          ! If /LOG, tell user
;  270        0266  3       THEN SIGNAL(set$_newfile);
;  271        0267  2       END;
;  272        0268  2
;  273        0269  2
;  274        0270  2   RETURN;
;  275        0271  1   END;


                                                                      .TITLE  SETACNTNG
                                                                      .IDENT  \V04-000\

                                                                      .PSECT  $PLIT$,NOWRT,NOEXE,2

                         00  00  45  4C  42  41  4E  45  00000 P.AAB:  .ASCII  \ENABLE\<0><0>
                                           010E0006  00008 P.AAA:  .LONG   17694726
                                           00000000' 0000C         .ADDRESS P.AAB
                         00  45  4C  42  41  53  49  44  00010 P.AAD:  .ASCII  \DISABLE\<0>
                                           010E0007  00018 P.AAC:  .LONG   17694727
                                           00000000' 0001C         .ADDRESS P.AAD
                         2C  53  53  45  43  4F  52  50  00020 P.AAF:  .ASCII  \PROCESS,\
                                           010E0008  00028 P.AAE:  .LONG   17694728
                                           00000000' 0002C         .ADDRESS P.AAF
                         00  00  2C  45  47  41  4D  49  00030 P.AAH:  .ASCII  \IMAGE,\<0><0>
                                           010E0006  00038 P.AAG:  .LONG   17694726
                                           00000000' 0003C         .ADDRESS P.AAH
         2C  45  56  49  54  43  41  52  45  54  4E  49  00040 P.AAJ:  .ASCII  \INTERACTIVE,\
                                           010E000C  0004C P.AAI:  .LONG   17694732
                                           00000000' 00050         .ADDRESS P.AAJ
00  2C  45  52  55  4C  49  41  46  5F  4E  49  47  4F  4C  00054 P.AAL:  .ASCII  \LOGIN_FAILURE,\<0><0>
                                                  00  00063
                                           010E000E  00064 P.AAK:  .LONG   17694734
                                           00000000' 00068         .ADDRESS P.AAL
     00  2C  53  53  45  43  4F  52  50  42  55  53  0006C P.AAN:  .ASCII  \SUBPROCESS,\<0>
                                           010E000B  00078 P.AAM:  .LONG   17694731
                                           00000000' 0007C         .ADDRESS P.AAN
     00  00  00  2C  44  45  48  43  41  54  45  44  00080 P.AAP:  .ASCII  \DETACHED,\<0><0><0>
                                           010E0009  0008C P.AAO:  .LONG   17694729
                                           00000000' 00090         .ADDRESS P.AAP
                         00  00  2C  48  43  54  41  42  00094 P.AAR:  .ASCII  \BATCH,\<0><0>
                                           010E0006  0009C P.AAQ:  .LONG   17694726
                                           00000000' 000A0         .ADDRESS P.AAR
                         2C  4B  52  4F  57  54  45  4E  000A4 P.AAT:  .ASCII  \NETWORK,\
                                           010E0008  000AC P.AAS:  .LONG   17694728
                                           00000000' 000B0         .ADDRESS P.AAT
                         00  00  2C  54  4E  49  52  50  000B4 P.AAV:  .ASCII  \PRINT,\<0><0>
                                           010E0006  000BC P.AAU:  .LONG   17694726
                                           00000000' 000C0         .ADDRESS P.AAV
                         2C  45  47  41  53  53  45  4D  000C4 P.AAX:  .ASCII  \MESSAGE,\
                                           010E0008  000CC P.AAW:  .LONG   17694728
                                           00000000' 000D0         .ADDRESS P.AAX
                                     00  47  4F  4C  000D4 P.AAZ:  .ASCII  \LOG\<0>
```

```
                                        010E0003  000D8 P.AAY:  .LONG   17694723                                          ;
                                        00000000' 000DC         .ADDRESS P.AAZ                                          ;
61  72  61  70  20  67  6E  69  74  6E  75  6F  63  63  61  000E0 P.ABB:  .ASCII  \accounting parameters\<0><0><0>          ;
                    00  00  00  73  72  65  74  65  6D  000EF                                                           ;
                                        010E0015  000F8 P.ABA:  .LONG   17694741                                        ;
                                        00000000' 000FC         .ADDRESS P.ABB                                          ;
                        45  4C  49  46  5F  57  45  4E  00100 P.ABD:  .ASCII  \NEW_FILE\                                 ;
                                        010E0008  00108 P.ABC:  .LONG   17694728                                        ;
                                        00000000' 0010C         .ADDRESS P.ABD                                          ;

                                                                .PSECT  $OWN$,NOEXE,2

                                00000000' 00000000' 00000 OPTION_STRING:
                                                                .ADDRESS P.AAA, P.AAC                                   ;
                                0000001A  00000017  00008 OPTION_CODE:
                                                                .LONG   23, 26                                          ;
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00010 ACC_NAME:
                                                                .ADDRESS P.AAE, P.AAG, P.AAI, P.AAK, P.AAM, -           ;
                    00000000' 00000000' 00000000' 00000000' 00028       P.AAO, P.AAQ, P.AAS, P.AAU, P.AAW               ;
00000020  00000010  00000008  00000004  00000002  00000001  00038 ACC_CODE:
                                                                .LONG   1, 2, 4, 8, 16, 32, 64, 128, 256, 512           ;
                    00000200  00000100  00000080  00000040  00050                                                      ;

                                                                .EXTRN  EXE$GL_ACMFLAGS
                                                                .EXTRN  STR$APPEND, CLI$GET_VALUE
                                                                .EXTRN  CLI$PRESENT, SET$_ACCENAB
                                                                .EXTRN  SET$_ACCDISAB, SET$_WRITEERR
                                                                .EXTRN  SET$_NEWFILE, SET$_NONEWFILE

                                                                .PSECT  $CODE$,NOWRT,2

                                        07FC 00000              .ENTRY  SET$ACCOUNTING, Save R2,R3,R4,R5,R6,R7,R8,-     ; 0146
                                                                        R9,R10                                          ;
                    5A 00000000V   EF   9E 00002                MOVAB   SEND_REQUEST, R10                           ;
                    59 00000000'   EF   9E 00009                MOVAB   P.AAY, R9                                   ;
                    58 00000000G   00   9E 00010                MOVAB   CLI$PRESENT, R8                             ;
                    57 00000000G   00   9E 00017                MOVAB   LIB$SIGNAL, R7                              ;
                    56 00000000'   EF   9E 0001E                MOVAB   OPTION_STRING, R6                           ;
                    5E                 1C   C2 00025            SUBL2   #28, SP                                     ;
                                       59   DD 00028            PUSHL   R9                                          ; 0173
                    68                 01   FB 0002A            CALLS   #1, CLI$PRESENT                             ;
                    55                 50   90 0002D            MOVB    R0, LOG                                     ;
                                       52   D4 00030            CLRL    I                                           ; 0181
                                     6642   DD 00032 1$:         PUSHL   OPTION_STRING[I]                            ; 0183
                    68                 01   FB 00035            CALLS   #1, CLI$PRESENT                             ;
                    4B                 50   E9 00038            BLBC    R0, 4$                                      ;
                                 18    AE   D4 0003B            CLRL    FLAGS                                       ; 0189
                                 18    AE   9F 0003E            PUSHAB  FLAGS                                       ; 0190
                                     6642   DD 00041            PUSHL   OPTION_STRING[I]                            ;
              00000000V  EF              02   FB 00044          CALLS   #2, PROCESS_REQUEST                         ;
                                 18    AE   D5 0004B            TSTL    FLAGS                                       ; 0195
                                       05   12 0004E            BNEQ    2$                                          ;
                                 08    AE   D4 00050            CLRL    BUFFER                                      ; 0196
                                       10   11 00053            BRB     3$                                          ;
                 08    AE 00020004    8F   D0 00055 2$:         MOVL    #131076, BUFFER                             ; 0205
                 0C    AE       18    AE   9E 0005D            MOVAB   FLAGS, BUFFER+4                             ; 0206
                                 10    AE   7C 00062            CLRQ    BUFFER+8                                    ; 0207
```

```
                        08   AE  9F  00065 3$:     PUSHAB  BUFFER                          ; 0214
                        08 A642  DD  00068         PUSHL   OPTION_CODE[I]                  ;
                    6A          FB  0006C         CALLS   #2, SEND_REQUEST                ;
                    54          D0  0006F         MOVL    R0, STATUS                      ;
                    11          E8  00072         BLBS    STATUS, 4$                      ;
                                DD  00075         PUSHL   STATUS                          ; 0220
                        20   A9  9F  00077         PUSHAB  P.ABA                           ; 0218
                                01  DD  0007A     PUSHL   #1                              ;
                 00000000G      8F  DD  0007C     PUSHL   #SET$_WRITEERR                  ;
                    67          04  FB  00082     CALLS   #4, LIB$SIGNAL                  ;
                                    04  00085     RET                                     ; 0217
                A8  52          01  F3  00086 4$:  AOBLEQ  #1, I, 1$                      ; 0181
                    47          55  E9  0008A     BLBC    LOG, 8$                         ; 0229
                    6E 020E0000 8F  D0  0008D     MOVL    #34471936, TYPES                ; 0235
                            04  AE  D4  00094     CLRL    TYPES+4                         ;
                    53 00000000G 00 D0  00097     MOVL    EXE$GL_ACMFLAGS, ACMFLAGS       ; 0236
                        52      D4  0009E         CLRL    I                               ; 0237
                OE  53          52  E1  000A0 5$:  BBC     I, ACMFLAGS, 6$                 ; 0238
                        10 A642  DD  000A4         PUSHL   ACC_NAME[I]                     ; 0239
                        04   AE  9F  000A8         PUSHAB  TYPES                           ;
             00000000G  00          02  FB  000AB     CALLS   #2, STR$APPEND              ;
                EA  52          09  F3  000B2 6$:  AOBLEQ  #9, I, 5$                      ; 0238
                                6E  B5  000B6     TSTW    TYPES                           ; 0240
                                0B  12  000B8     BNEQ    7$                              ;
                 00000000G      8F  DD  000BA     PUSHL   #SET$_ACCDISAB                  ; 0241
                    67          01  FB  000C0     CALLS   #1, LIB$SIGNAL                  ;
                                0F  11  000C3     BRB     8$                              ;
                                6E  B7  000C5 7$:  DECW    TYPES                           ; 0244
                                5E  DD  000C7     PUSHL   SP                              ; 0245
                                01  DD  000C9     PUSHL   #1                              ;
                 00000000G      8F  DD  000CB     PUSHL   #SET$_ACCENAB                   ;
                    67          03  FB  000D1     CALLS   #3, LIB$SIGNAL                  ;
                            30  A9  9F  000D4 8$:  PUSHAB  P.ABC                           ; 0254
                    68          01  FB  000D7     CALLS   #1, CLI$PRESENT                 ;
                    37          50  E9  000DA     BLBC    R0, 10$                         ;
                08  AE 00680000 8F  D0  000DD     MOVL    #6815744, BUFFER                ; 0257
                            0C  AE  7C  000E5     CLRQ    BUFFER+4                        ; 0258
                            14  AE  D4  000E8     CLRL    BUFFER+12                       ; 0260
                            08  AE  9F  000EB     PUSHAB  BUFFER                          ; 0261
                            08  A6  DD  000EE     PUSHL   OPTION_CODE                     ;
                    6A          02  FB  000F1     CALLS   #2, SEND_REQUEST                ;
                    54          50  D0  000F4     MOVL    R0, STATUS                      ;
                    0E          54  E8  000F7     BLBS    STATUS, 9$                      ;
                                54  DD  000FA     PUSHL   STATUS                          ; 0264
                                7E  D4  000FC     CLRL    -(SP)                           ; 0263
                 00000000G      8F  DD  000FE     PUSHL   #SET$_NONEWFILE                 ;
                    67          03  FB  00104     CALLS   #3, LIB$SIGNAL                  ;
                                    04  00107     RET                                     ;
                    09          55  E9  00108 9$:  BLBC    LOG, 10$                        ; 0265
                 00000000G      8F  DD  0010B     PUSHL   #SET$_NEWFILE                   ; 0266
                    67          01  FB  00111     CALLS   #1, LIB$SIGNAL                  ;
                                    04  00114 10$:  RET                                   ; 0271

; Routine Size:  277 bytes,    Routine Base:  $CODE$ + 0000
```

```
0272  1 ROUTINE process_request (option, flags) : NOVALUE =
0273  2 BEGIN
0274  2
0275  2 !++
0276  2 ! Functional description
0277  2 !
0278  2 !       This routine collects the qualifiers to indicate what types
0279  2 !       of accounting data to enable or disable.  A corresponding bit
0280  2 !       is set in the FLAGS word.
0281  2 !
0282  2 ! Inputs
0283  2 !
0284  2 !       OPTION - address of descriptor for operation (ENABLE or DISABLE)
0285  2 !       FLAGS  - longword bitmask describing what was set
0286  2 !
0287  2 ! Outputs
0288  2 !       FLAGS  - bitmask set to indicate what was requested
0289  2 !
0290  2 !----
0291  2
0292  2 LOCAL
0293  2     desc : $BBLOCK[dsc$c_s_bln];          ! CLI value descriptor
0294  2
0295  2 $init_dyndesc(desc);                     ! Make the descriptor dynamic
0296  2
0297  2 !
0298  2 ! For each accounting category specified by the user, OR in the corresponding
0299  2 ! bitmask to the flags longword.
0300  2 !
0301  2 WHILE cli$get_value(.option, desc) DO
0302  3     BEGIN
0303  3     INCR i FROM 0 TO 9 DO
0304  4         BEGIN
0305  4         BIND name = .acc_name[.i] : VECTOR;
0306  4         IF CH$EQL(.desc[dsc$w_length], .desc[dsc$a_pointer],
0307  4                 .desc[dsc$w_length], .name[1])
0308  4         THEN EXITLOOP (.flags = ..flags OR .acc_code[.i]);
0309  3         END;
0310  2     END;
0311  2
0312  2 RETURN;
0313  1 END;
```

```
                                 001C 00000 PROCESS_REQUEST:
                                             .WORD   Save R2,R3,R4                  ; 0272
               5E        04  C2 00002        SUBL2   #4, SP
             020E0000    8F  DD 00005        PUSHL   #34471936                      ; 0295
                   04    AE  D4 0000B        CLRL    DESC+4
                         5E  DD 0000E 1$:    PUSHL   SP                             ; 0301
                   04    AC  DD 00010        PUSHL   OPTION                         ;
     00000000G    00     02  FB 00013        CALLS   #2, CLI$GET_VALUE              ;
                  23     50  E9 0001A        BLBC    R0, 4$                         ;
                         54  D4 0001D        CLRL    I                              ; 0306
```

```
                   50 00000000'EF44  D0 0001F 2$:   MOVL    ACC_NAME[I], R0                  ; 0305
   04  B0     04  BE            6E  29 00027        CMPC3   DESC, @DESC+4, @4(R0)            ; 0306
                                0B  12 0002D        BNEQ    3$                               ;
              08  BC 00000000'EF44  C8 0002F        BISL2   ACC_CODE[I], @FLAGS              ; 0308
                                D4  11 00038        BRB     1$                               ;
       E1         54            09  F3 0003A 3$:    AOBLEQ  #9, I, 2$                        ; 0303
                                CE  11 0003E        BRB     1$                               ; 0301
                                    04 00040 4$:    RET                                      ; 0313

; Routine Size:  65 bytes,    Routine Base:  $CODE$ + 0115
```

```
;   320          0314  1 ROUTINE send_request (function, buffer) =
;   321          0315  2 BEGIN
;   322          0316  2
;   323          0317  2 !++
;   324          0318  2 !
;   325          0319  2 !       This routine sends the request to the accounting manager, and
;   326          0320  2 !       obtains a status return.
;   327          0321  2 !
;   328          0322  2 ! Inputs
;   329          0323  2 !       FUNCTION - function to perform - start, stop accounting
;   330          0324  2 !       BUFFER - message buffer to send to the acc. manager.
;   331          0325  2 !
;   332          0326  2 ! Outputs
;   333          0327  2 !       Final status is returned.
;   334          0328  2 !
;   335          0329  2 !----
;   336          0330  2
;   337          0331  2 LOCAL
;   338          0332  2     status,
;   339          0333  2     iosb : VECTOR[2];
;   340          0334  2
;   341          0335  2 !
;   342          0336  2 ! Send the request to the accounting manager.
;   343          0337  2 !
;   344        P 0338  2 status = $SNDJBCW(FUNC    = .function,
;   345        P 0339  2                   ITMLST = .buffer,
;   346          0340  2                   IOSB   = iosb);
;   347          0341  2 IF .status
;   348          0342  2 THEN status = .iosb[0];
;   349          0343  2
;   350          0344  2 RETURN .status;                         ! Return the final status
;   351          0345  1 END;


                                                .EXTRN  SYS$SNDJBCW

                                0000 00000 SEND_REQUEST:
                                                .WORD   Save nothing                    ; 0314
                            5E                  SUBL2   #8, SP                          ;
                                    7E  7C 00005        CLRQ    -(SP)                   ; 0340
                                08  AE  9F 00007        PUSHAB  IOSB                    ;
                                08  AC  DD 0000A        PUSHL   BUFFER                  ;
                                    7E  D4 0000D        CLRL    -(SP)                   ;
                                04  AC  DD 0000F        PUSHL   FUNCTION                ;
                                    7E  D4 00012        CLRL    -(SP)                   ;
                    00000000G   00      FB 00014        CALLS   #7, SYS$SNDJBCW         ;
                                03      E9 0001B        BLBC    STATUS, 1$              ; 0341
                                50      D0 0001E        MOVL    IOSB, STATUS            ; 0342
                                        04 00021 1$:    RET                             ; 0345

; Routine Size:  34 bytes,    Routine Base:  $CODE$ + 0156
```

```
;  353          0346  1 END
;  354          0347  0 ELUDOM


                                                        .EXTRN  LIB$SIGNAL

;                               PSECT SUMMARY

;       Name                    Bytes                   Attributes

;  $PLIT$                        272  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $OWN$                          96  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                        376  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        26         0       581        00:01.0
```

```
;                                   COMMAND QUALIFIERS

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

; Size:          376 code + 368 data bytes
; Run Time:          00:09.3
; Elapsed Time:      00:30.3
; Lines/CPU Min:      2248
; Lexemes/CPU-Min: 16056
; Memory Used:   100 pages
; Compilation Complete
```

SETACNTNG
LIS

RENAMEMSG
LIS

RENAME
LIS

RUNDET
LIS

RUNMSG
LIS

RUNCVTUIC
LIS

SET
LIS

0052 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

SETDEVICE.LIS

SETDEVS.LIS

SETCLUSUB.LIS

SETDIR.LIS

SETACT.LIS

SETAUDIT.LIS

SETCLUSTR.LIS