PPPPPPPPPPPP RRRRRRRRRRR TTTTTTTTTTTTTTT SSSSSSSSSSS MMM MMM BBBBBBBBBBB
PPPPPPPPPPPP RRRRRRRRRRR TTTTTTTTTTTTTTT SSSSSSSSSSS MMM MMM BBBBBBBBBBB
PPPPPPPPPPPP RRRRRRRRRRR TTTTTTTTTTTTTTT SSSSSSSSSSS MMM MMM BBBBBBBBBBB
PPP PPP RRR RRR TTT SSS MMMMMM MMMMMM BBB BBB
PPP PPP RRR RRR TTT SSS MMMMMM MMMMMM BBB BBB
PPP PPP RRR RRR TTT SSS MMMMMM MMMMMM BBB BBB
PPP PPP RRR RRR TTT SSS MMM MMM MMM BBB BBB
PPP PPP RRR RRR TTT SSS MMM MMM MMM BBB BBB
PPP PPP RRR RRR TTT SSS MMM MMM MMM BBB BBB
PPPPPPPPPPPP RRRRRRRRRRR TTT SSSSSSSSS MMM MMM BBBBBBBBBBB
PPPPPPPPPPPP RRRRRRRRRRR TTT SSSSSSSSS MMM MMM BBBBBBBBBBB
PPPPPPPPPPPP RRRRRRRRRRR TTT SSSSSSSSS MMM MMM BBBBBBBBBBB
PPP RRR RRR TTT SSS MMM MMM BBB BBB
PPP RRR RRR TTT SSS MMM MMM BBB BBB
PPP RRR RRR TTT SSS MMM MMM BBB BBB
PPP RRR RRR TTT SSS MMM MMM BBB BBB
PPP RRR RRR TTT SSS MMM MMM BBB BBB
PPP RRR RRR TTT SSSSSSSSSSS MMM MMM BBBBBBBBBBB
PPP RRR RRR TTT SSSSSSSSSSS MMM MMM BBBBBBBBBBB
PPP RRR RRR TTT SSSSSSSSSSS MMM MMM BBBBBBBBBBB
PPP RRR RRR TTT SSSSSSSSSSS MMM MMM BBBBBBBBBBB

```
MM      MM  EEEEEEEEEE  MM      MM    000000    RRRRRRRR    YY      YY
MM      MM  EEEEEEEEEE  MM      MM    000000    RRRRRRRR    YY      YY
MMMM  MMMM  EE          MMMM  MMMM  00      00  RR      RR  YY      YY
MMMM  MMMM  EE          MMMM  MMMM  00      00  RR      RR  YY      YY
MM  MM  MM  EE          MM  MM  MM  00      00  RR      RR    YY  YY
MM  MM  MM  EE          MM  MM  MM  00      00  RR      RR    YY  YY
MM      MM  EEEEEEEE    MM      MM  00      00  RRRRRRRR        YY
MM      MM  EEEEEEEE    MM      MM  00      00  RRRRRRRR        YY
MM      MM  EE          MM      MM  00      00  RR  RR          YY
MM      MM  EE          MM      MM  00      00  RR  RR          YY
MM      MM  EE          MM      MM  00      00  RR    RR        YY          ....
MM      MM  EE          MM      MM  00      00  RR    RR        YY          ....
MM      MM  EEEEEEEEEE  MM      MM    000000    RR      RR      YY          ....
MM      MM  EEEEEEEEEE  MM      MM    000000    RR      RR      YY          ....

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 MEMORY (%TITLE 'Symbiont Services -- Memory routines'
;     2        0002  0                  IDENT = 'V04-000'
;     3        0003  0                  ADDRESSING_MODE (EXTERNAL = GENERAL)
;     4        0004  0                  ) =
;     5        0005  1 BEGIN
;     6        0006  1
;     7        0007  1 !
;     8        0008  1 !**************************************************************************
;     9        0009  1 !*                                                                        *
;    10        0010  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                               *
;    11        0011  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                 *
;    12        0012  1 !*  ALL RIGHTS RESERVED.                                                   *
;    13        0013  1 !*                                                                        *
;    14        0014  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    15        0015  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    16        0016  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    17        0017  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    18        0018  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    19        0019  1 !*  TRANSFERRED.                                                           *
;    20        0020  1 !*                                                                        *
;    21        0021  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    22        0022  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    23        0023  1 !*  CORPORATION.                                                           *
;    24        0024  1 !*                                                                        *
;    25        0025  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;    26        0026  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                *
;    27        0027  1 !*                                                                        *
;    28        0028  1 !*                                                                        *
;    29        0029  1 !**************************************************************************
;    30        0030  1 !
;    31        0031  1
;    32        0032  1 !++
;    33        0033  1 ! FACILITY:
;    34        0034  1 !       Symbiont Services
;    35        0035  1 !
;    36        0036  1 ! ABSTRACT:
;    37        0037  1 !       This module provides routines to allocate, deallocate and initialize
;    38        0038  1 !       memory structures.
;    39        0039  1 !
;    40        0040  1 ! ENVIRONMENT:
;    41        0041  1 !       VAX/VMS user mode.
;    42        0042  1 !--
;    43        0043  1 !
;    44        0044  1 ! AUTHOR: Greg Robert, CREATION DATE: 26-Apr-1983
;    45        0045  1 !
;    46        0046  1 ! MODIFIED BY:
;    47        0047  1 !
;    48        0048  1 !       3B-007  RRB3007         Rowland R. Bradley      18-Jul-1984
;    49        0049  1 !               Insert COPY_R_DX to PSM$READ_ITEM_DX. This insures proper
;    50        0050  1 !               copy of longword items to users descriptor.  Change message
;    51        0051  1 !               PSM$_NOMOREITEMS to SMB$_NOMOREITEMS, PSM$_INVSTRLEV to
;    52        0052  1 !               SMB$_INVSTRLEV.
;    53        0053  1 !
;    54        0054  1 !       3B-006  GRR3006         Gregory R. Robert       16-May-1984
;    55        0055  1 !               Reset current PAGE and START_PAGE on SCB reset.
;    56        0056  1 !
;    57        0057  1 !       3B-005  GRR3005         Gregory R. Robert       29-Apr-1984
```

```
;    58          0058  1 !                   FT2 bugfixes.
;    59          0059  1 !
;    60          0060  1 !           3B-004  RRB3004         Rowland R. Bradley      27-Apr-1984
;    61          0061  1 !                   Add support for the new page header routine.  Initialize
;    62          0062  1 !                   the page header dynamic descriptor.
;    63          0063  1 !
;    64          0064  1 !           3B-003  GRR3003         Gregory R. Robert       23-Aug-1983
;    65          0065  1 !                   Bugfixes, page_setup_modules, form_setup_modules,
;    66          0066  1 !                   sheet_feed, symbiont initiated pause_task and stop_stream,
;    67          0067  1 !                   hangup code, read and write item services
;    68          0068  1 !
;    69          0069  1 !           3B-002  GRR3002         Gregory R. Robert       03-Aug-1983
;    70          0070  1 !                   Rewrite for new design.
;    71          0071  1 !
;    72          0072  1 !           3B-001  GRR4001         Gregory R. Robert       24-Jul-1983
;    73          0073  1 !                   Created new module
;    74          0074  1 !
;    75          0075  1 !**
```

```
;    77          0076  1
;    78          0077  1 LIBRARY 'SYS$LIBRARY:LIB';
;    79          0078  1 REQUIRE 'LIB$:SMBDEF';
;    80          0570  1 REQUIRE 'SRC$:SMBREQ';
;    81          1027  1
;    82          1028  1 EXTERNAL
;    83          1029  1         PSM$MIT          : BLOCKVECTOR[,MIT_S_MIT, BYTE],
;    84          1030  1         PSM$XLATE_8BIT   : VECTOR [,BYTE]
;    85          1031  1         ;
;    86          1032  1
;    87          1033  1 EXTERNAL ROUTINE
;    88          1034  1         SMB$READ_MESSAGE_ITEM    ,
;    89          1035  1         EXPAND_CONDITION_VECTOR
;    90          1036  1         ;
;    91          1037  1
;    92          1038  1 FORWARD ROUTINE
;    93          1039  1         PSM$ALLOCATE_DSB         : NOVALUE,
;    94          1040  1         PSM$ALLOCATE_IOB         : NOVALUE,
;    95          1041  1         PSM$DEALLOCATE_DSB       : NOVALUE,
;    96          1042  1         PSM$DEALLOCATE_IOB       : NOVALUE,
;    97          1043  1         PSM$ALLOCATE_SCB         ,
;    98          1044  1         PSM$INITIALIZE_SCB       ,
;    99          1045  1         PSM$READ_ITEM_DX         ,
;   100          1046  1         PSM$READ_ITEM_R          ,
;   101          1047  1         PSM$RESET_SCB            ,
;   102          1048  1         PSM$WRITE_ITEM_DX        ,
;   103          1049  1         PSM$WRITE_ITEM_R         ,
;   104          1050  1         PSM$UPDATE_SCB           ,
;   105          1051  1         ALLOCATE_MEMORY
;   106          1052  1         ;
;   107          1053  1
;   108          1054  1 OWN
;   109          1055  1         FREE_DSB_QUEUE: VECTOR [2],     ! Dynamic string block queue header
;   110          1056  1         FREE_IOB_QUEUE: VECTOR [2]      ! IO block queue header
;   111          1057  1         ;
```

```
;  113          1058  1 %SBTTL 'ALLOCATE_DSB'
;  114          1059  1 ! Functional Description:
;  115          1060  1 !               Allocates a dynanic string block.
;  116          1061  1 !
;  117          1062  1 ! Formal Parameters:
;  118          1063  1 !               DSB     : address of longword to receive allocated DSB address
;  119          1064  1 !               BYTES   : number of bytes to be reserved in dynamic string desc.
;  120          1065  1 !
;  121          1066  1 ! Implicit Inputs:
;  122          1067  1 !               none
;  123          1068  1 !
;  124          1069  1 ! Implicit Outputs:
;  125          1070  1 !               none
;  126          1071  1 !
;  127          1072  1 ! Returned Value:
;  128          1073  1 !               none
;  129          1074  1 !
;  130          1075  1 ! Side Effects:
;  131          1076  1 !               none
;  132          1077  1 !--
;  133          1078  1 GLOBAL ROUTINE PSM$ALLOCATE_DSB (
;  134          1079  1         DSB     : REF $LONGWORD,
;  135          1080  1         BYTES   : REF $WORD
;  136          1081  1         )       : NOVALUE =
;  137          1082  2 BEGIN
;  138          1083  2         PARAMETER_INDEX_ (DSB, BYTES);
;  139          1084  2
;  140          1085  2
;  141          1086  2 IF .FREE_DSB_QUEUE[0] EQL 0
;  142          1087  2 THEN
;  143          1088  2     INIT_QUEUE_HEADER_ (FREE_DSB_QUEUE);
;  144          1089  2
;  145          1090  2 ! Dequeue a dynamic string block
;  146          1091  2 !
;  147          1092  2 WHILE REMOVE_HEAD_ (DSB[], FREE_DSB_QUEUE)      ! True if empty                                    ; R
;  148          1093  2 DO
;  149          1094  3     BEGIN
;  150          1095  3     LOCAL PAGE;
;  151          1096  3
;  152          1097  3     ! Allocate a page of DSB's.
;  153          1098  3     !
;  154          1099  3     $ASSUME (DSB_S_DSB * 32, EQL, 512)
;  155          1100  3     PAGE = ALLOCATE_MEMORY (512);
;  156          1101  3     CH$FILL (0, 512, .PAGE);
;  157          1102  3
;  158          1103  3     ! Initialize and place them in queue
;  159          1104  3     !
;  160          1105  3     INCRA PTR FROM .PAGE TO .PAGE + 511 BY DSB_S_DSB
;  161          1106  3     DO
;  162          1107  4         BEGIN
;  163          1108  4         MAP PTR: REF $BBLOCK;
;  164          1109  4         INIT_DYN_DESC  (PTR[DSB_Q_DESC]);
;  165          1110  4         INSERT_TAIL_ (PTR[DSB_Q_QLINKS], FREE_DSB_QUEUE);
;  166          1111  3         END;
;  167          1112  2     END;
;  168          1113  2
;  169          1114  2 DSB[] = .DSB[] - $BYTEOFFSET (DSB_Q_QLINKS);
```

```
;  170          1115  2
;  171          1116  2 IF PARAMETER_PRESENT_ (BYTES)
;  172          1117  2 THEN
;  173          1118  2     SIGNAL_IF_ERROR_ (STR$GET1_DX (BYTES[], $BBLOCK[.DSB[], DSB_Q_DESC[]));
;  174          1119  2
;  175          1120  1 END;


                                                .TITLE  MEMORY Symbiont Services -- Memory routines
                                                .IDENT  \V04-000\

                                                .PSECT  DATA,NOEXE,2

                                        00000 FREE_DSB_QUEUE:
                                                .BLKB   8
                                        00008 FREE_IOB_QUEUE:
                                                .BLKB   8

                                                .EXTRN  BAS$EDIT, LBR$CLOSE
                                                .EXTRN  LBR$GET_RECORD, LBR$INI_CONTROL
                                                .EXTRN  LBR$LOOKUP_KEY, LBR$OPEN
                                                .EXTRN  LBR$RET_RMSSTV, LBR$SET_LOCATE
                                                .EXTRN  LIB$TRIM_FILESPEC
                                                .EXTRN  LIB$GET_VM, LIB$FREE_VM
                                                .EXTRN  STR$ANALYZE_SDESC
                                                .EXTRN  STR$ANALYZE_SDESC_R1
                                                .EXTRN  STR$APPEND, STR$CONCAT
                                                .EXTRN  STR$COPY_DX, STR$COPY_R
                                                .EXTRN  STR$FREE1_DX, STR$FREE1_DX_R4
                                                .EXTRN  STR$GET1_DX, STR$LEFT
                                                .EXTRN  STR$PREFIX, STR$RIGHT
                                                .EXTRN  PSM$S_HANGUP_DISPATCH_ENTRY
                                                .EXTRN  PSM$_BUFFEROVF, PSM$_EOF
                                                .EXTRN  PSM$_ESCAPE, PSM$_FLUSH
                                                .EXTRN  PSM$_FUNNOTSUP, PSM$_INVITMCOD
                                                .EXTRN  PSM$_INVVMSOSC, PSM$_MODNOTFND
                                                .EXTRN  PSM$_NEWPAGE, PSM$_NOFILEID
                                                .EXTRN  PSM$_OSCTOOLON, PSM$_PENDING
                                                .EXTRN  PSM$_SUSPEND, PSM$_TOOMANYLEV
                                                .EXTRN  SMB$_INVSTMNBR, SMB$_INVSTRLEV
                                                .EXTRN  SMB$_NOMOREITEMS
                                                .EXTRN  PSM$RMIT, PSM$XLATE_8BIT
                                                .EXTRN  SMB$READ_MESSAGE_ITEM
                                                .EXTRN  EXPAND_CONDITION_VECTOR

                                                .PSECT  CODE,NOWRT,2

                                          00FC 00000            .ENTRY  PSM$ALLOCATE_DSB, Save R2,R3,R4,R5,R6,R7    ; 1078
        57      0000'   CF   9E 00002          MOVAB   FREE_DSB_QUEUE, R7                                          ;
                        67   D5 00007          TSTL    FREE_DSB_QUEUE                                              ; 1086
                        07   12 00009          BNEQ    1$                                                          ;
        67              67   9E 0000B          MOVAB   FREE_DSB_QUEUE, FREE_DSB_QUEUE                              ; 1088
  04    A7              67   9E 0000E          MOVAB   FREE_DSB_QUEUE, FREE_DSB_QUEUE+4                            ;
  04    BC        00    B7   0F 00012 1$:      REMQUE  @FREE_DSB_QUEUE, @DSB                                       ; 1092
                        3B   1C 00017          BVC     4$                                                          ;
        7E      0200    8F   3C 00019          MOVZWL  #512, -(SP)                                                 ; 1100
0000V   CF              01   FB 0001E          CALLS   #1, ALLOCATE_MEMORY                                         ;
```

```
                                56              50  D0 00023          MOVL    R0, PAGE
    0200    8F          00      6E              00  2C 00026          MOVC5   #0, (SP), #0, #512, (PAGE)                  ;1101
                                                66     0002D
                                52      01FF    C6  9E 0002E          MOVAB   511(R6), R2                                 ;1105
                                50              56  D0 00033          MOVL    PAGE, PTR
                                                15  11 00036          BRB     3$
                                51        08    A0  9E 00038 2$:      MOVAB   8(PTR), R1                                  ;1109
                                61 020E0000     8F  D0 0003C          MOVL    #34471936, (R1)
                                          04    A1  D4 00043          CLRL    4(R1)
                            04  B7              60  0E 00046          INSQUE  (PTR), @FREE_DSB_QUEUE+4                    ;1110
                                50              10  C0 0004A          ADDL2   #16, PTR                                    ;1105
                                52              50  D1 0004D 3$:      CMPL    PTR, R2
                                                E6  1B 00050          BLEQU   2$
                                                BE  11 00052          BRB     1$
                                                                                                                        ;1092
                                02              6C  91 00054 4$:      CMPB    (AP), #2                                    ;1116
                                                23  1F 00057          BLSSU   5$
                                          08    AC  D5 00059          TSTL    8(AP)
                                                1E  13 0005C          BEQL    5$
                        7E      04  BC              08  C1 0005E          ADDL3   #8, @DSB, -(SP)                         ;1118
                                          08    AC  DD 00063          PUSHL   BYTES
                    00000000G   00              02  FB 00066          CALLS   #2, STR$GET1_DX
                                52              50  D0 0006D          MOVL    R0, STATUS
                                09              52  E8 00070          BLBS    STATUS, 5$
                                                52  DD 00073          PUSHL   STATUS
                    00000000G   00              01  FB 00075          CALLS   #1, LIB$SIGNAL
                                                    04 0007C 5$:      RET                                                 ;1120

; Routine Size:  125 bytes,    Routine Base:  CODE + 0000
```

```
;  177          1121  1 %SBTTL 'ALLOCATE_IOB'
;  178          1122  1 ! Functional Description:
;  179          1123  1 !               Allocate an Input Output control Block.
;  180          1124  1 !
;  181          1125  1 ! Formal Parameters:
;  182          1126  1 !               IOB - address of a longword to receive address of allocated IOB
;  183          1127  1 !               BYTES - number of bytes to be allocated in buffer
;  184          1128  1 !
;  185          1129  1 ! Implicit Inputs:
;  186          1130  1 !               none
;  187          1131  1 !
;  188          1132  1 ! Implicit Outputs:
;  189          1133  1 !               none
;  190          1134  1 !
;  191          1135  1 ! Returned Value:
;  192          1136  1 !               none
;  193          1137  1 !
;  194          1138  1 ! Side Effects:
;  195          1139  1 !               none
;  196          1140  1 !--
;  197          1141  1 GLOBAL ROUTINE PSM$ALLOCATE_IOB (
;  198          1142  1         IOB     : REF $LONGWORD,
;  199          1143  1         BYTES   : REF $WORD
;  200          1144  1         )       : NOVALUE =
;  201          1145  2 BEGIN
;  202          1146  2
;  203          1147  2 BUILTIN
;  204          1148  2         NULLPARAMETER
;  205          1149  2         ;
;  206          1150  2
;  207          1151  2 IF .FREE_IOB_QUEUE[0] EQL 0
;  208          1152  2 THEN
;  209          1153  2     INIT_QUEUE_HEADER_ (FREE_IOB_QUEUE);
;  210          1154  2
;  211          1155  2
;  212          1156  2 ! Dequeue a dynamic string block
;  213          1157  2 !
;  214          1158  2 WHILE REMOVE_HEAD_ (IOB[], FREE_IOB_QUEUE)      ! True if empty
;  215          1159  2 DO
;  216          1160  3     BEGIN
;  217          1161  3     LOCAL MEMORY;
;  218          1162  3
;  219          1163  3     ! Allocate 20 IOB's.
;  220          1164  3     !
;  221          1165  3     MEMORY = ALLOCATE_MEMORY (20 * IOB_S_IOB);
;  222          1166  3     CH$FILL (0, 20 * IOB_S_IOB, .MEMORY);
;  223          1167  3
;  224          1168  3     ! Initialize and place them in queue
;  225          1169  3     !
;  226          1170  3     INCRA PTR FROM .MEMORY TO .MEMORY + (20 * IOB_S_IOB - 1) BY IOB_S_IOB
;  227          1171  3     DO
;  228          1172  4         BEGIN
;  229          1173  4         MAP PTR: REF $BBLOCK;
;  230          1174  4         PTR[IOB_B_TYPE] = PSM$K_STRUCTURE_IOB;
;  231          1175  4         PTR[IOB_B_LEVEL] = SMBMSG$K_STRUCTURE_LEVEL;
;  232          1176  4         PTR[IOB_W_SIZE] = IOB_S_IOB;
;  233          1177  4         INIT_DYN_DESC_ (PTR[IOB_Q_BUFFER]);
```

```
;  234          1178  4          INSERT_TAIL_ (PTR[IOB_Q_QLINKS], FREE_IOB_QUEUE);
;  235          1179  3          END;
;  236          1180  2      END;
;  237          1181  2
;  238          1182  2 IOB[] = .IOB[] - $BYTEOFFSET (IOB_Q_QLINKS);
;  239          1183  2
;  240          1184  2 IF NOT NULLPARAMETER (2)
;  241          1185  2 THEN
;  242          1186  2      SIGNAL_IF_ERROR_ (STR$GET1_DX (BYTES[], $BBLOCK[.IOB[], IOB_Q_BUFFER]));
;  243          1187  2
;  244          1188  1 END;


                                                00FC 00000          .ENTRY  PSM$ALLOCATE_IOB, Save R2,R3,R4,R5,R6,R7    ; 1141
                                    57      0000' CF   9E 00002     MOVAB   FREE_IOB_QUEUE, R7                          ;
                                                  67   D5 00007     TSTL    FREE_IOB_QUEUE                              ; 1151
                                                  07   12 00009     BNEQ    1$                                          ;
                                    67            67   9E 0000B     MOVAB   FREE_IOB_QUEUE, FREE_IOB_QUEUE             ; 1153
                                04  A7            67   9E 0000E     MOVAB   FREE_IOB_QUEUE, FREE_IOB_QUEUE+4           ;
                                04  BC      00    B7   0F 00012 1$:  REMQUE  @FREE_IOB_QUEUE, @IOB                       ; 1158
                                                  44   1C 00017     BVC     4$                                          ;
                                    7E      05A0  8F   3C 00019     MOVZWL  #1440, -(SP)                                ; 1165
                              0000V CF            01   FB 0001E     CALLS   #1, ALLOCATE_MEMORY                         ;
                                    56            50   D0 00023     MOVL    R0, MEMORY                                  ;
    05A0    8F          00        6E            00   2C 00026     MOVC5   #0, (SP), #0, #1440, (MEMORY)               ; 1166
                                                  66     0002D                                                          ;
                                    52      059F  C6   9E 0002E     MOVAB   1439(R6), R2                                ; 1170
                                    50            56   D0 00033     MOVL    MEMORY, PTR                                 ;
                                                  1E   11 00036     BRB     3$                                          ;
                                08  A0 00480102    8F   D0 00038 2$:  MOVL    #4718850, 8(PTR)                        ; 1174
                                    51      1C    A0   9E 00040     MOVAB   28(PTR), R1                                 ; 1177
                                    61 020E0000   8F   D0 00044     MOVL    #34471936, (R1)                             ;
                                           04     A1   D4 0004B     CLRL    4(R1)                                       ;
                                04  B7            60   0E 0004E     INSQUE  (PTR), @FREE_IOB_QUEUE+4                    ; 1178
                                    50      48    A0   9E 00052     MOVAB   72(R0), PTR                                 ; 1170
                                    52            50   D1 00056 3$:  CMPL    PTR, R2                                     ;
                                                  DD   1B 00059     BLEQU   2$                                          ;
                                                  B5   11 0005B     BRB     1$                                          ; 1158
                                    02            6C   91 0005D 4$:  CMPB    (AP), #2                                ; 1184
                                                  23   1F 00060     BLSSU   5$                                          ;
                                           08     AC   D5 00062     TSTL    8(AP)                                       ;
                                                  1E   13 00065     BEQL    5$                                          ;
                          7E        04    BC            1C   C1 00067     ADDL3   #28, @IOB, -(SP)                        ; 1186
                                           08     AC   DD 0006C     PUSHL   BYTES                                       ;
                             00000000G    00            02   FB 0006F     CALLS   #2, STR$GET1_DX                     ;
                                    52            50   D0 00076     MOVL    R0, STATUS                                  ;
                                    09            52   E8 00079     BLBS    STATUS, 5$                                  ;
                                                  52   DD 0007C     PUSHL   STATUS                                      ;
                             00000000G    00            01   FB 0007E     CALLS   #1, LIB$SIGNAL                      ;
                                                     04 00085 5$:    RET                                                 ; 1188

; Routine Size:  134 bytes,    Routine Base:  CODE + 007D
```

```
;  246          1189  1 %SBTTL 'DEALLOCATE_DSB'
;  247          1190  1 !++
;  248          1191  1 ! Functional Description:
;  249          1192  1 !               Deallocate a Dynamic String Block
;  250          1193  1 !
;  251          1194  1 ! Formal Parameters:
;  252          1195  1 !               DSB -- address of block to be deallocated
;  253          1196  1 !
;  254          1197  1 ! Implicit Inputs:
;  255          1198  1 !               none
;  256          1199  1 !
;  257          1200  1 ! Implicit Outputs:
;  258          1201  1 !               none
;  259          1202  1 !
;  260          1203  1 ! Returned Value:
;  261          1204  1 !               none
;  262          1205  1 !
;  263          1206  1 ! Side Effects:
;  264          1207  1 !               none                                                    ; 1
;  265          1208  1 !--
;  266          1209  1 GLOBAL ROUTINE PSM$DEALLOCATE_DSB (
;  267          1210  1         DSB     : REF $BBLOCK
;  268          1211  1         )       : NOVALUE =
;  269          1212  2 BEGIN
;  270          1213  2 SIGNAL_IF_ERROR_ (STR$FREE1_DX (DSB[DSB_Q_DESC]));
;  271          1214  2
;  272          1215  2 INSERT_TAIL_ (DSB[DSB_Q_QLINKS], FREE_DSB_QUEUE);
;  273          1216  2
;  274          1217  1 END;


                                        0004 00000         .ENTRY  PSM$DEALLOCATE_DSB, Save R2                     ; 1208
                        7E      04      AC          08  C1 00002         ADDL3   #8, DSB, -(SP)                          ; 1213
                00000000G       00              01  FB 00007         CALLS   #1, STR$FREE1_DX                        ;
                                52              50  D0 0000E         MOVL    R0, STATUS                              ;
                                09              52  E8 00011         BLBS    STATUS, 1$                              ;
                                                52  DD 00014         PUSHL   STATUS                                  ;
                00000000G       00              01  FB 00016         CALLS   #1, LIB$SIGNAL                          ;
                    0000'       DF      04      BC  0E 0001D 1$:      INSQUE  @DSB, @FREE_DSB_QUEUE+4                 ; 1215
                                                    04 00023         RET                                             ; 1217

; Routine Size:  36 bytes,    Routine Base:  CODE + 0103
```

```
;  276          1218  1 %SBTTL 'DEALLOCATE_IOB'
;  277          1219  1 ! Functional Description:
;  278          1220  1 !                   Deallocates an Input/Output Block
;  279          1221  1 !
;  280          1222  1 ! Formal Parameters:
;  281          1223  1 !                   IOB - address of block to be deallocated
;  282          1224  1 !
;  283          1225  1 ! Implicit Inputs:
;  284          1226  1 !                   none
;  285          1227  1 !
;  286          1228  1 ! Implicit Outputs:
;  287          1229  1 !                   none
;  288          1230  1 !
;  289          1231  1 ! Returned Value:
;  290          1232  1 !                   none
;  291          1233  1 !
;  292          1234  1 ! Side Effects:
;  293          1235  1 !                   none
;  294          1236  1 !--
;  295          1237  1 GLOBAL ROUTINE PSM$DEALLOCATE_IOB (
;  296          1238  1           IOB     : REF $BBLOCK
;  297          1239  1           )       : NOVALUE =
;  298          1240  2 BEGIN
;  299          1241  2
;  300          1242  2 SIGNAL_IF_ERROR_ (STR$FREE1_DX (IOB[IOB_Q_BUFFER]));
;  301          1243  2
;  302          1244  2 INSERT_TAIL_ (IOB[IOB_Q_QLINKS], FREE_IOB_QUEUE);
;  303          1245  2
;  304          1246  1 END;


                                        0004 00000         .ENTRY  PSM$DEALLOCATE_IOB, Save R2                     ;  1237
                7E      04   AC          1C   C1 00002     ADDL3   #28, IOB, -(SP)                                 ;  1242
                   00000000G 00          01   FB 0C007     CALLS   #1, STR$FREE1_DX                                ;
                             52          50   D0 0000E     MOVL    R0, STATUS                                      ;
                             09          52   E8 00011     BLBS    STATUS, 1$                                      ;
                                         52   DD 00014     PUSHL   STATUS                                          ;
                   00000000G 00          01   FB 00016     CALLS   #1, LIB$SIGNAL                                  ;
                      0000' DF      04   BC   0E 0001D 1$:  INSQUE  @IOB, @FREE_IOB_QUEUE+4                        ;  1244
                                         04   00023         RET                                                     ;  1246

; Routine Size:  36 bytes,    Routine Base:  CODE  + 0127
```

```
;   306          1247  1 %SBTTL 'ALLOCATE_SCB'
;   307          1248  1 ! Functional Description:
;   308          1249  1 !                   Allocates a Stream Control Block
;   309          1250  1 !
;   310          1251  1 ! Formal Parameters:
;   311          1252  1 !                   SCBADR - address of a longword to receive address of new SCB
;   312          1253  1 !
;   313          1254  1 ! Implicit Inputs:
;   314          1255  1 !                   none
;   315          1256  1 !
;   316          1257  1 ! Implicit Outputs:
;   317          1258  1 !                   none
;   318          1259  1 !
;   319          1260  1 ! Returned Value:
;   320          1261  1 !                   none
;   321          1262  1 !
;   322          1263  1 ! Side Effects:
;   323          1264  1 !                   none
;   324          1265  1 !--
;   325          1266  1 GLOBAL ROUTINE PSM$ALLOCATE_SCB (
;   326          1267  1           SCBADR  : REF $LONGWORD
;   327          1268  1           )       =
;   328          1269  2 BEGIN
;   329          1270  2
;   330          1271  2 EXTERNAL
;   331          1272  2           PSM$GL_USER_CTX
;   332          1273  2           ;
;   333          1274  2
;   334          1275  2 LOCAL
;   335          1276  2           SCB             : REF $BBLOCK,
;   336          1277  2           SCB_SIZE        : INITIAL (PSM$S_SCB)                                          ; 1
;   337          1278  2           ;
;   338          1279  2
;   339          1280  2 SCB_SIZE = .SCB_SIZE + .PSM$GL_USER_CTX;
;   340          1281  2
;   341          1282  2 RETURN_IF_ERROR_ (LIB$GET_VM (SCB_SIZE, SCB));
;   342          1283  2
;   343          1284  2 CH$FILL (0, .SCB_SIZE, .SCB);
;   344          1285  2
;   345          1286  2 SCB[PSM$B_TYPE] = PSM$K_STRUCTURE_SCB;
;   346          1287  2 SCB[PSM$B_LEVEL] = SMBMSG$K_STRUCTURE_LEVEL;
;   347          1288  2 SCB[PSM$W_SIZE] = PSM$S_SCB;
;   348          1289  2
;   349          1290  2 SCBADR[] = .SCB;
;   350          1291  2
;   351          1292  2 SS$_NORMAL
;   352          1293  2
;   353          1294  1 END;


                                                                .EXTRN  PSM$GL_USER_CTX

                                                 007C 00000     .ENTRY  PSM$ALLOCATE_SCB, Save R2,R3,R4,R5,R6       ; 1266
                                      5E         08  C2 00002   SUBL2   #8, SP                                      ;
                                04    AE     02D4  8F  3C 00005   MOVZWL  #724, SCB_SIZE                           ; 1269
                                04    AE 00000000G  00  C0 0000B   ADDL2   PSM$GL_USER_CTX, SCB_SIZE              ; 1280
```

```
                                                  5E  DD 00013          PUSHL   SP                                          ; 1282
                                              08  AE  9F 00015          PUSHAB  SCB_SIZE                                    ;
                                 00000000G    00  02  FB 00018          CALLS   #2, LIB$GET_VM                              ;
                                              19  50  E9 0001F          BLBC    STATUS, 1$                                  ;
                                              56  6E  D0 00022          MOVL    SCB, R6                                     ; 1284
      04    AE              00                6E  00  2C 00025          MOVC5   #0, (SP), #0, SCB_SIZE, (R6)                ;
                                                    66    0002B
                                    08        A6  8F  D0 0002C          MOVL    #47448323, 8(R6)                            ; 1286
                                    04        BC  56  D0 00034          MOVL    R6, @SCBADR                                 ; 1290
                                              50  01  D0 00038          MOVL    #1, R0                                      ; 1294
                                                  04 0003B 1$:          RET                                                 ;

; Routine Size:  60 bytes,    Routine Base:  CODE + 014B
```

```
;  355     1295  1 %SBTTL 'INITIALIZE_SCB'
;  356     1296  1 ! Functional Description:
;  357     1297  1 !               Initializes an SCB at START_STREAM time.
;  358     1298  1 !
;  359     1299  1 ! Formal Parameters:
;  360     1300  1 !               SCB     - address of SCB to be initialized
;  361     1301  1 !
;  362     1302  1 ! Implicit Inputs:
;  363     1303  1 !               PSM$MIT - message item table containing information
;  364     1304  1 !                         about message items in SCB.
;  365     1305  1 !
;  366     1306  1 ! Implicit Outputs:
;  367     1307  1 !               none
;  368     1308  1 !
;  369     1309  1 ! Returned Value:
;  370     1310  1 !               none
;  371     1311  1 !
;  372     1312  1 ! Side Effects:
;  373     1313  1 !               none
;  374     1314  1 !--
;  375     1315  1 GLOBAL ROUTINE PSM$INITIALIZE_SCB (
;  376     1316  1         SCB     : REF $BBLOCK
;  377     1317  1         )       =
;  378     1318  2 BEGIN
;  379     1319  2
;  380     1320  2 LOCAL
;  381     1321  2         IOB :   REF $BBLOCK                                     ; F
;  382     1322  2         ;
;  383     1323  2
;  384     1324  2
;  385     1325  2 IF .SCB[PSM$B_LEVEL] NEQU SMBMSG$K_STRUCTURE_LEVEL
;  386     1326  2 OR .SCB[PSM$B_TYPE]  NEQU PSM$K_STRUCTURE_SCB
;  387     1327  2 THEN
;  388     1328  2     RETURN SMB$_INVSTRLEV;
;  389     1329  2
;  390     1330  2
;  391     1331  2 DECR ITEM_CODE FROM SMBMSG$K_MAX_ITEM_CODE TO 0
;  392     1332  2 DO
;  393     1333  2     IF .PSM$MIT[.ITEM_CODE, MIT_B_TYPE] EQL MIT_K_DYNAMIC
;  394     1334  2     THEN
;  395     1335  2         INIT_DYN_DESC_ (.SCB + .PSM$MIT[.ITEM_CODE, MIT_W_OFFSET]);
;  396     1336  2
;  397     1337  2
;  398     1338  2 SCB_SIZE_ (ACCOUNTING_DATA) = PSM$S_ACCOUNTING_AREA;
;  399     1339  2 SCB_ADDR_ (ACCOUNTING_DATA) = SCB[PSM$T_ACCOUNTING_AREA];
;  400     1340  2
;  401     1341  2
;  402     1342  2 SCB_SIZE_ (CONDITION_VECTOR) = PSM$S_CONDITION_AREA;
;  403     1343  2 SCB_ADDR_ (CONDITION_VECTOR) = SCB[PSM$T_CONDITION_AREA];
;  404     1344  2
;  405     1345  2
;  406     1346  2 INIT_DYN_DESC_ (SCB[PSM$Q_CONDITION_TEXT]);
;  407     1347  2 INIT_DYN_DESC_ (SCB[PSM$Q_USER_BUFFER]);
;  408     1348  2 STR$GET1_DX (UPLIT WORD (PSM$K_DEFBUFSIZ), SCB[PSM$Q_USER_BUFFER]);
;  409     1349  2 INIT_DYN_DESC_ (SCB[PSM$Q_MODULE_LIST]);
;  410     1350  2 INIT_DYN_DESC_ (SCB[PSM$Q_MODULE_NAME]);
;  411     1351  2 INIT_DYN_DESC_ (SCB[PSM$Q_PAGE_HEADER]);
```

```
;  412          1352  2 INIT_DYN_DESC_ (SCB[PSM$Q_SEARCH_CONTEXT]);
;  413          1353  2
;  414          1354  2
;  415          1355  2 INIT_QUEUE_HEADER_ (SCB[PSM$Q_CHECKPOINT_QUEUE]);
;  416          1356  2 INIT_QUEUE_HEADER_ (SCB[PSM$Q_INPUT_QUEUE]);
;  417          1357  2
;  418          1358  2
;  419          1359  2 ! Cause the first longword of the SCB to point to itself.
;  420          1360  2 !
;  421          1361  2 $ASSUME ($BYTEOFFSET (PSM$Q_QLINKS), EQL, 0)
;  422          1362  2 INIT_QUEUE_HEADER_ (SCB[PSM$Q_QLINKS]);
;  423          1363  2
;  424          1364  2 SS$_NORMAL
;  425          1365  2
;  426          1366  1 END;


                                                00187          .BLKB   1
                                          0200  00188 P.AAA:   .WORD   512                                          ;

                                                0004 00000     .ENTRY  PSM$INITIALIZE_SCB, Save R2                    ; 1315
                                          52    04 AC  D0 00002  MOVL   SCB, R2                                      ; 1325
                                          01    09 A2  91 00006  CMPB   9(R2), #1                                    ;
                                                06     12 0000A  BNEQ   1$                                           ;
                                          03    08 A2  91 0000C  CMPB   8(R2), #3                                    ; 1326
                                                08     13 00010  BEQL   2$                                           ;
                                          50 00000000G 8F  D0 00012 1$:  MOVL   #SMB$_INVSTRLEV, R0                   ; 1328
                                                04     00019     RET                                                  ;
                                          51          39  D0 0001A 2$:  MOVL   #57, ITEM_CODE                       ; 1331
                                             00000000G00041  DF 0001D 3$:  PUSHAL  PSM$MIT[ITEM_CODE]               ; 1333
                                       FF    8F        9E  91 00024     CMPB   @(SP)+, #-1                          ;
                                                17     12 00028     BNEQ   4$                                       ;
                                             00000000G00041  DF 0002A     PUSHAL  PSM$MIT+2[ITEM_CODE]               ; 1335
                                          50          9E  3C 00031     MOVZWL @(SP)+, R0                           ;
                                          50          52  C0 00034     ADDL2  R2, R0                               ;
                                          60 020E0000   8F  D0 00037     MOVL   #34471936, (R0)                      ;
                                                04      A0  D4 0003E     CLRL   4(R0)                                ;
                                          D9          51  F4 00041 4$:  SOBGEQ ITEM_CODE, 3$                       ; 1333
                                          50          14  A2 9E 00044     MOVAB  20(R2), R0                           ; 1338
                                          60          10  B0 00048     MOVW   #16, (R0)                            ;
                                       04    A0     027E  C2 9E 0004B     MOVAB  638(R2), 4(R0)                       ; 1339
                                          50          44  A2 9E 00051     MOVAB  68(R2), R0                           ; 1342
                                          60          14  B0 00055     MOVW   #20, (R0)                            ;
                                       04    A0     028E  C2 9E 00058     MOVAB  654(R2), 4(R0)                       ; 1343
                                          50        0198  C2 9E 0005E     MOVAB  408(R2), R0                          ; 1346
                                          60 020E0000   8F  D0 00063     MOVL   #34471936, (R0)                      ;
                                                04      A0  D4 0006A     CLRL   4(R0)                                ;
                                          50        023C  C2 9E 0006D     MOVAB  572(R2), R0                          ; 1347
                                          60 020E0000   8F  D0 00072     MOVL   #34471936, (R0)                      ;
                                                04      A0  D4 00079     CLRL   4(R0)                                ;
                                                        50  DD 0007C     PUSHL  R0                                   ; 1348
                                                FF7C    CF  9F 0007E     PUSHAB P.AAA                                ;
                                    00000000G   00          02 FB 00082  CALLS  #2, STR$GET1_DX                      ;
                                          50        01CC  C2 9E 00089     MOVAB  460(R2), R0                          ; 1349
```

```
                60 020E0000   8F  D0 0008E        MOVL    #34471936, (R0)                 ;
                         04   A0  D4 00095        CLRL    4(R0)                           ;
                50       01D4 C2  9E 00098        MOVAB   468(R2), R0                     ; 1350
                60 020E0000   8F  D0 0009D        MOVL    #34471936, (R0)                 ;
                         04   A0  D4 000A4        CLRL    4(R0)                           ;
                50       01F0 C2  9E 000A7        MOVAB   496(R2), R0                     ; 1351
                60 020E0000   8F  D0 000AC        MOVL    #34471936, (R0)                 ;
                         04   A0  D4 000B3        CLRL    4(R0)                           ;
                50       0210 C2  9E 000B6        MOVAB   528(R2), R0                     ; 1352
                60 020E0000   8F  D0 000BB        MOVL    #34471936, (R0)                 ;
                         04   A0  D4 000C2        CLRL    4(R0)                           ;
                50       017C C2  9E 000C5        MOVAB   380(R2), R0                     ; 1355
                60             50  D0 000CA        MOVL    R0, (R0)                        ;
             04 A0             50  D0 000CD        MOVL    R0, 4(R0)                       ;
                50       0184 C2  9E 000D1        MOVAB   388(R2), R0                     ; 1356
                60             50  D0 000D6        MOVL    R0, (R0)                        ;
             04 A0             50  D0 000D9        MOVL    R0, 4(R0)                       ;
                62             52  D0 000DD        MOVL    R2, (R2)                        ; 1362
             04 A2             52  D0 000E0        MOVL    R2, 4(R2)                       ;
                50             01  D0 000E4        MOVL    #1, R0                          ; 1366
                                   04 000E7        RET                                     ;

; Routine Size:  232 bytes,    Routine Base:  CODE + 018A
```

```
;  428          1367  1 %SBTTL 'READ_ITEM_DX'
;  429          1368  1 ! Functional Description:
;  430          1369  1 !               Reads a message item from the SCB into a descriptor.
;  431          1370  1 !
;  432          1371  1 ! Formal Parameters:
;  433          1372  1 !               SMB_CONTEXT     - address of the SCB
;  434          1373  1 !               ITEM_CODE       - item to be read
;  435          1374  1 !               DESCRIPTOR      - address of output buffer descriptor
;  436          1375  1 !
;  437          1376  1 ! Implicit Inputs:
;  438          1377  1 !               none
;  439          1378  1 !
;  440          1379  1 ! Implicit Outputs:
;  441          1380  1 !               none
;  442          1381  1 !
;  443          1382  1 ! Returned Value:
;  444          1383  1 !               none
;  445          1384  1 !
;  446          1385  1 ! Side Effects:
;  447          1386  1 !               none
;  448          1387  1 !--
;  449          1388  1 GLOBAL ROUTINE PSM$READ_ITEM_DX (
;  450          1389  1         SMB_CONTEXT     : REF $LONGWORD,
;  451          1390  1         ITEM_CODE       : REF $LONGWORD,
;  452          1391  1         DESCRIPTOR      : REF $BBLOCK
;  453          1392  1         ) =
;  454          1393  2 BEGIN
;  455          1394  2
;  456          1395  2 LOCAL   ITEM_ADR;
;  457          1396  2
;  458          1397  2 IF .ITEM_CODE[] GEQU SMBMSG$K_MAX_ITEM_CODE THEN RETURN PSM$_INVITMCOD;
;  459          1398  2
;  460          1399  2 ITEM_ADR = .SMB_CONTEXT[] + .PSM$MIT[.ITEM_CODE[], MIT_W_OFFSET];
;  461          1400  2
;  462          1401  2 CASE .PSM$MIT[.ITEM_CODE[], MIT_B_TYPE] FROM MIT_K_STATIC TO -1 OF
;  463          1402  2     SET
;  464          1403  2
;  465          1404  2     [MIT_K_DYNAMIC, MIT_K_STATIC]:
;  466          1405  2         COPY_DX_DX_ (.ITEM_ADR, .DESCRIPTOR);
;  467          1406  2
;  468          1407  2     [OUTRANGE]:
;  469          1408  3         BEGIN
;  470          1409  3         LOCAL SIZE;
;  471          1410  3         SIZE = .PSM$MIT[.ITEM_CODE[], MIT_B_TYPE];      ! Copy needs a ref.
;  472          1411  3
;  473        P 1412  3         COPY_R_DX_ ( SIZE,                              ! Table item size
;  474        P 1413  3                      .ITEM_ADR,                         ! Table item address
;  475          1414  4                      .DESCRIPTOR )                      ! User item desc
;  476          1415  4
;  477          1416  2         END;
;  478          1417  2     TES;
;  479          1418  2
;  480          1419  2 SS$_NORMAL
;  481          1420  2
;  482          1421  1 END;
```

```
                                    0004 00000          .ENTRY  PSM$READ_ITEM_DX, Save R2                       ; 1388
                                5E      04  C2 00002       SUBL2   #4, SP                                          ;
                                50      08  BC  D0 00005   MOVL    @ITEM_CODE, R0                                  ; 1397
                                39          50  D1 00009   CMPL    R0, #57                                         ;
                                            08  1F 0000C   BLSSU   1$                                              ;
                                50 00000000G    8F  D0 0000E   MOVL    #PSM$_INVITMCOD, R0                         ;
                                                04  00015   RET                                                     ;
                                   00000000G00040   DF  00016 1$:   PUSHAL  PSM$MIT+2[R0]                          ; 1399
                                51          9E  3C  0001D   MOVZWL  @(SP)+, ITEM_ADR                             ;
                                51      04  BC  C0  00020   ADDL2   @SMB_CONTEXT, ITEM_ADR                       ;
                                   00000000G00040   DF  00024   PUSHAL  PSM$MIT[R0]                               ; 1401
                                50          9E  98  0002B   CVTBL   @(SP)+, R0                                    ;
                        01      FE  8F          50  8F  0002E   CASEB   R0, #-2, #1                               ;
                                  0018      0018      00033 2$:   .WORD   3$-2$,-                                   ;
                                                                          3$-2$                                     ;
                                6E          50  D0  00037   MOVL    R0, SIZE                                    ; 1410
                                            51  DD  0003A   PUSHL   ITEM_ADR                                     ; 1414
                                        04  AE  9F  0003C   PUSHAB  SIZE                                         ;
                                        0C  AC  DD  0003F   PUSHL   DESCRIPTOR                                  ;
                    00000000G   00          03  FB  00042   CALLS   #3, STR$COPY_R                              ;
                                            0C  11  00049   BRB     4$                                          ;
                                            51  DD  0004B 3$:   PUSHL   ITEM_ADR                                ; 1405
                                        0C  AC  DD  0004D   PUSHL   DESCRIPTOR                                  ;
                    00000000G   00          02  FB  00050   CALLS   #2, STR$COPY_DX                             ;
                                52          50  D0  00057 4$:   MOVL    R0, STATUS                              ;
                                09          52  E8  0005A   BLBS    STATUS, 5$                                  ;
                                            52  DD  0005D   PUSHL   STATUS                                      ;
                    00000000G   00          01  FB  0005F   CALLS   #1, LIB$SIGNAL                             ;
                                50          01  D0  00066 5$:   MOVL    #1, R0                                  ; 1421
                                                04  00069   RET                                                     ;

; Routine Size:  106 bytes,    Routine Base:  CODE + 0272
```

```
;  484          1422  1 %SBTTL 'READ_ITEM_R'
;  485          1423  1 !++
;  486          1424  1 ! Functional Description:
;  487          1425  1 !               NOT SUPPORTED IN V4.0.
;  488          1426  1 !
;  489          1427  1 ! Formal Parameters:
;  490          1428  1 !               ?desc
;  491          1429  1 ! Implicit Inputs:
;  492          1430  1 !               none
;  493          1431  1 !
;  494          1432  1 ! Implicit Outputs:
;  495          1433  1 !               none
;  496          1434  1 !
;  497          1435  1 ! Returned Value:
;  498          1436  1 !               none
;  499          1437  1 !
;  500          1438  1 ! Side Effects:
;  501          1439  1 !               none
;  502          1440  1 !--
;  503          1441  1 GLOBAL ROUTINE PSM$READ_ITEM_R (
;  504          1442  1         SMB_CONTEXT     : REF $LONGWORD,
;  505          1443  1         ITEM_CODE       : REF $LONGWORD,
;  506          1444  1         BUFSIZ          : REF $WORD,
;  507          1445  1         BUFADR          : REF VECTOR [,BYTE]
;  508          1446  2         ) =
;  509          1447  2 BEGIN
;  510          1448  2 LOCAL USER_DESCRIPTOR : VECTOR [2];
;  511          1449  2
;  512          1450  2 USER_DESCRIPTOR [0] = .BUFSIZ[];
;  513          1451  2 USER_DESCRIPTOR [1] = .BUFADR;
;  514          1452  2
;  515          1453  2 RETURN PSM$READ_ITEM_DX (.SMB_CONTEXT, .ITEM_CODE, USER_DESCRIPTOR);
;  516          1454  2
;  517          1455  1 END;


                                        0000 00000      .ENTRY  PSM$READ_ITEM_R, Save nothing                   ; 1440
                                5E          04  C2 00002    SUBL2   #4, SP                                          ;
                                7E      0C  BC  3C 00005    MOVZWL  @BUFSIZ, USER_DESCRIPTOR                        ; 1450
                            04  AE      10  AC  D0 00009    MOVL    BUFADR, USER_DESCRIPTOR+4                       ; 1451
                                            5E  DD 0000E    PUSHL   SP                                              ; 1453
                                7E      04  AC  7D 00010    MOVQ    SMB_CONTEXT, -(SP)                              ;
                          FF7D  CF          03  FB 00014    CALLS   #3, PSM$READ_ITEM_DX                            ;
                                                04 00019    RET                                                     ; 1455

; Routine Size:  26 bytes,    Routine Base:  CODE + 02DC
```

```
;  519          1456  1 %SBTTL 'RESET_SCB'
;  520          1457  1 ! Functional Description:
;  521          1458  1 !                 Resets an SCB for starting a new task
;  522          1459  1 !
;  523          1460  1 ! Formal Parameters:
;  524          1461  1 !                 SCB     : SCB address
;  525          1462  1 !
;  526          1463  1 ! Implicit Inputs:
;  527          1464  1 !                 PSM$MIT - message item table containing information
;  528          1465  1 !                           about message items in SCB.
;  529          1466  1 !
;  530          1467  1 ! Implicit Outputs:
;  531          1468  1 !                 none
;  532          1469  1 !
;  533          1470  1 ! Returned Value:
;  534          1471  1 !                 none
;  535          1472  1 !
;  536          1473  1 ! Side Effects:
;  537          1474  1 !                 none
;  538          1475  1 !--
;  539          1476  1 GLOBAL ROUTINE PSM$RESET_SCB (
;  540          1477  1         SCB     : REF $BBLOCK
;  541          1478  1         )       =
;  542          1479  2 BEGIN
;  543          1480  2
;  544          1481  2 DECR ITEM_CODE FROM SMBMSG$K_MAX_ITEM_CODE TO 0
;  545          1482  2 DO
;  546          1483  3     BEGIN
;  547          1484  3     BIND ITEM = .SCB + .PSM$MIT[.ITEM_CODE, MIT_W_OFFSET];
;  548          1485  3     IF .PSM$MIT[.ITEM_CODE, MIT_V_RESET]
;  549          1486  3     THEN
;  550          1487  3         CASE .PSM$MIT[.ITEM_CODE, MIT_B_TYPE] FROM MIT_K_STATIC TO -1 OF
;  551          1488  3             SET
;  552          1489  3
;  553          1490  3             [MIT_K_DYNAMIC]:
;  554          1491  3                 IF .DESC_SIZE_ (ITEM) NEQ 0
;  555          1492  3                 THEN
;  556          1493  3                     STR$FREE1_DX (ITEM);
;  557          1494  3
;  558          1495  3             [MIT_K_STATIC]:
;  559          1496  3                 CH$FILL (0, .DESC_SIZE_ (ITEM), .DESC_ADDR_ (ITEM));
;  560          1497  3
;  561          1498  3             [OUTRANGE]:
;  562          1499  3                 CH$FILL (0, .PSM$MIT[.ITEM_CODE, MIT_B_TYPE], ITEM);
;  563          1500  3
;  564          1501  3             TES;
;  565          1502  2     END;
;  566          1503  2
;  567          1504  2
;  568          1505  2 STR$FREE1_DX (SCB[PSM$Q_CONDITION_TEXT]);
;  569          1506  2 STR$FREE1_DX (SCB[PSM$Q_MODULE_LIST]);
;  570          1507  2 STR$FREE1_DX (SCB[PSM$Q_MODULE_NAME]);
;  571          1508  2
;  572          1509  2
;  573          1510  2 CH$FILL (0, PSM$S_ACCOUNTING_AREA, SCB[PSM$T_ACCOUNTING_AREA]);
;  574          1511  2 CH$FILL (0, PSM$S_CONDITION_AREA, SCB[PSM$T_CONDITION_AREA]);
;  575          1512  2
```

```
;  576          1513  2
;  577          1514  2 SCB[PSM$L_PAGE]                    = 0;
;  578          1515  2 SCB[PSM$L_PRINT_FLAGS]            = 0;
;  579          1516  2 SCB[PSM$L_START_PAGE]             = 0;
;  580          1517  2 SCB[PSM$L_STOP_PAGE]              = 0;
;  581          1518  2 SCB[PSM$L_TASK_FLAGS]             = 0;
;  582          1519  2
;  583          1520  2 SCB[PSM$L_T_MARGIN]              = 0;
;  584          1521  2 SCB[PSM$L_L_MARGIN]              = 0;
;  585          1522  2
;  586          1523  2 CLEAR_QUAD_ (SCB[PSM$Q_ITEM_FLAGS]);
;  587          1524  2
;  588          1525  2 SCB[PSM$L_KEEP_ALIVE]            = -1;
;  589          1526  2 SCB[PSM$L_RECORD_NUMBER]         = 1;
;  590          1527  2 SCB[PSM$L_SERVICE_STATUS]        = SS$_NORMAL;
;  591          1528  2 SCB[PSM$A_XLATE_TABLE]           = PSM$XLATE_8BIT;
;  592          1529  2
;  593          1530  2
;  594          1531  2 ! Release any outstanding IO control block
;  595          1532  2 !
;  596          1533  2 IF .SCB[PSM$A_IOB] NEQ 0
;  597          1534  2 THEN
;  598          1535  3     BEGIN
;  599        P 1536  3     INSERT_TAIL_ (.SCB[PSM$A_IOB] + $BYTEOFFSET(IOB_Q_QLINKS),
;  600          1537  3         SCB[PSM$Q_BUFFER_QUEUE]);
;  601          1538  3     SCB[PSM$A_IOB] = 0;
;  602          1539  2     END;
;  603          1540  2
;  604          1541  2
;  605          1542  2 ! Empty the checkpoint queue
;  606          1543  2 !
;  607          1544  3 BEGIN
;  608          1545  3 LOCAL PTR: REF $BBLOCK;
;  609          1546  3 UNTIL REMOVE_HEAD_ (PTR, SCB[PSM$Q_CHECKPOINT_QUEUE])    ! True when empty
;  610          1547  3 DO
;  611          1548  3     PSM$DEALLOCATE_DSB (.PTR - $BYTEOFFSET (DSB_Q_QLINKS));
;  612          1549  3 SCB[PSM$B_CHECKPOINT_DEPTH] = 0;
;  613          1550  2 END;
;  614          1551  2
;  615          1552  2
;  616          1553  2 ! Empty the nested input queue
;  617          1554  2 !
;  618          1555  3 BEGIN
;  619          1556  3 LOCAL PTR: REF $BBLOCK;
;  620          1557  3 UNTIL REMOVE_HEAD_ (PTR, SCB[PSM$Q_INPUT_QUEUE])        ! True when empty
;  621          1558  3 DO
;  622          1559  3     PSM$DEALLOCATE_DSB (.PTR - $BYTEOFFSET (DSB_Q_QLINKS));
;  623          1560  3 SCB[PSM$B_INPUT_DEPTH] = 0;
;  624          1561  2 END;
;  625          1562  2
;  626          1563  2
;  627          1564  2 SS$_NORMAL
;  628          1565  2
;  629          1566  1 END;
```

```
                                                07FC 00000          .ENTRY  PSM$RESET_SCB, Save R2,R3,R4,R5,R6,R7,R8,-      ; 1476
                                                                            R9,R10
                                    5A 00000000G   00   9E 00002    MOVAB   PSM$MIT+2, R10                                  ;
                                    59 00000000G   00   9E 00009    MOVAB   STR$FREE1_DX, R9                                ;
                                    56         04     AC   D0 00010    MOVL    SCB, R6                                         ; 1484
                                    57               39   D0 00014    MOVL    #57, ITEM_CODE                                  ;
                                                  6A47   DF 00017 1$: PUSHAL  PSM$MIT+2[ITEM_CODE]                            ;
                                    50               9E   3C 0001A    MOVZWL  @(SP)+, R0                                      ;
                                58  56               50   C1 0001D    ADDL3   R0, R6, R8                                      ;
                                    2A             FF AA47   E9 00021    BLBC    PSM$MIT+1[ITEM_CODE], 5$                 ; 1485
                                                   FE AA47   DF 00026    PUSHAL  PSM$MIT[ITEM_CODE]                       ; 1487
                                    50               9E   98 0002A    CVTBL   @(SP)+, R0                                      ;
                                01      FE       8F         50   8F 0002D    CASEB   R0, #-2, #1                             ;
                                          000C           0017        00032 2$: .WORD   4$-2$,-                                 ;
                                                                               3$-2$                                        ;
                50              00         6E               00   2C 00036    MOVC5   #0, (SP), #0, R0, (R8)           ; 1499
                                                            68        0003B                                                   ;
                                                            12   11 0003C    BRB     5$                                      ;
                                                            68   B5 0003E 3$: TSTW    (R8)                             ; 1491
                                                            0E   13 00040    BEQL    5$                                      ;
                                                            58   DD 00042    PUSHL   R8                               ; 1493
                                    69                      01   FB 00044    CALLS   #1, STR$FREE1_DX                        ;
                                                            07   11 00047    BRB     5$                               ; 1491
                68              00         6E               00   2C 00049 4$: MOVC5   #0, (SP), #0, (R8), @4(R8)       ; 1496
                                                      04    B8        0004E                                                   ;
                                    r4               57   F4 00050 5$: SOBGEQ  ITEM_CODE, 1$                           ; 1481
                                                0198  C6   9F 00053    PUSHAB  408(R6)                                  ; 1505
                                    69               01   FB 00057    CALLS   #1, STR$FREE1_DX                        ;
                                                01CC  C6   9F 0005A    PUSHAB  460(R6)                                  ; 1506
                                    69               01   FB 0005E    CALLS   #1, STR$FREE1_DX                        ;
                                                01D4  C6   9F 00061    PUSHAB  468(R6)                                  ; 1507
                                    69               01   FB 00065    CALLS   #1, STR$FREE1_DX                        ;
                10              00         6E               00   2C 00068    MOVC5   #0, (SP), #0, #16, 638(R6)       ; 1510
                                                027E  C6         0006D                                                   ;
                14              00         6E               00   2C 00070    MOVC5   #0, (SP), #0, #20, 654(R6)       ; 1511
                                                028E  C6         00075                                                   ;
                                                01EC  C6   D4 00078    CLRL    492(R6,                                  ; 1514
                                                0204  C6   D4 0007C    CLRL    516(R6)                                  ; 1515
                                                0224  C6   7C 00080    CLRQ    548(R6)                                  ; 1516
                                                  10  A6   D4 00084    CLRL    16(R6)                                   ; 1518
                                                0230  C6   D4 00087    CLRL    560(R6)                                  ; 1520
                                                01BC  C6   D4 0008B    CLRL    444(R6)                                  ; 1521
                                    50          01B0  C6   9E 0008F    MOVAB   432(R6), R0                              ; 1523
                                                      60   7C 00094    CLRQ    (R0)                                     ;
                                          01B8  C6         01   CE 00096    MNEGL   #1, 440(R6)                              ; 1525
                                          026C  C6         01   D0 0009B    MOVL    #1, 620(R6)                              ; 1526
                                          0220  C6         01   D0 000A0    MOVL    #1, 544(R6)                              ; 1527
                                          0244  C6 00000000G   00   9E 000A5    MOVAB   PSM$XLATE_8BIT, 580(R6)          ; 1528
                                                50    01AC   C6   D0 000AE    MOVL    428(R6), R0                              ; 1533
                                                             0D   13 000B3    BEQL    6$                                       ;
                                          0178  D6               60   0E 000B5    INSQUE  (R0), @376(R6)                   ; 1537
                                                50         04    AC   D0 000BA    MOVL    SCB, R0                          ; 1538
                                                      01AC  C0   D4 000BE    CLRL    428(R0)                                  ;
                                                50         04    AC   D0 000C2 6$: MOVL    SCB, R0                          ; 1546
                                                52    017C   D0   0F 000C6    REMQUE  @380(R0), PTR                            ;
```

```
                                    09  1D 000CB            BVS     7$                                          ;
                                    52  DD 000CD            PUSHL   PTR                                         ; 1548
                          FD39  CF      01  FB 000CF            CALLS   #1, PSM$DEALLOCATE_DSB                  ;
                                    EC  11 000D4            BRB     6$                                          ;
                          50        04  AC  D0 000D6 7$:     MOVL    SCB, R0                                     ; 1549
                                02A2  C0  94 000DA            CLRB    674(R0)                                     ;
                          50        04  AC  D0 000DE 8$:     MOVL    SCB, R0                                     ; 1557
                          52      0184  D0  0F 000E2            REMQUE  @388(R0), PTR                           ;
                                    09  1D 000E7            BVS     9$                                          ;
                                    52  DD 000E9            PUSHL   PTR                                         ; 1559
                          FD1D  CF      01  FB 000EB            CALLS   #1, PSM$DEALLOCATE_DSB                  ;
                                    EC  11 000F0            BRB     8$                                          ;
                          50        04  AC  D0 000F2 9$:     MOVL    SCB, R0                                     ; 1560
                                02A5  C0  94 000F6            CLRB    677(R0)                                     ;
                          50            01  D0 000FA            MOVL    #1, R0                                  ; 1566
                                        04 000FD            RET                                                 ;

; Routine Size:  254 bytes,    Routine Base:  CODE + 02F6
```

```
;  631          1567  1 %SBTTL 'UPDATE_SCB'
;  632          1568  1 ! Functional Description:
;  633          1569  1 !               Updates an SCB with messages items
;  634          1570  1 !
;  635          1571  1 ! Formal Parameters:
;  636          1572  1 !               SCB     - address of SCB to be updated
;  637          1573  1 !               MESSAGE - address of descriptor of message
;  638          1574  1 !
;  639          1575  1 ! Implicit Inputs:
;  640          1576  1 !               none
;  641          1577  1 !
;  642          1578  1 ! Implicit Outputs:
;  643          1579  1 !               none
;  644          1580  1 !
;  645          1581  1 ! Returned Value:
;  646          1582  1 !               none
;  647          1583  1 !
;  648          1584  1 ! Side Effects:
;  649          1585  1 !               none
;  650          1586  1 !--
;  651          1587  1 GLOBAL ROUTINE PSM$UPDATE_SCB (
;  652          1588  1         SCB     : REF $BBLOCK,
;  653          1589  1         MESSAGE : REF VECTOR
;  654          1590  1         )       =
;  655          1591  2 BEGIN
;  656          1592  2
;  657          1593  2 BUILTIN
;  658          1594  2         SP
;  659          1595  2         ;
;  660          1596  2
;  661          1597  2 LOCAL
;  662          1598  2         STATUS_1,
;  663          1599  2         CONTEXT         : INITIAL (0),
;  664          1600  2         ITEM            : INITIAL (0),
;  665          1601  2         DESC            : $DYNAMIC_DESC
;  666          1602  2         ;
;  667          1603  2
;  668          1604  2 SCB[PSM$L_REQUEST_CONTROL] = 0;
;  669          1605  2
;  670          1606  3 WHILE (STATUS_1 = SMB$READ_MESSAGE_ITEM (.MESSAGE, CONTEXT, ITEM, DESC))
;  671          1607  2 DO
;  672          1608  3     BEGIN
;  673          1609  3     BITVECTOR [SCB[PSM$Q_ITEM_FLAGS], .ITEM] = 1;
;  674          1610  3     PSM$WRITE_ITEM_DX (SCB, ITEM, DESC);
;  675          1611  2     END;
;  676          1612  2
;  677          1613  2 IF .STATUS_1 NEQ SMB$_NOMOREITEMS
;  678          1614  2 THEN
;  679          1615  2     RETURN .STATUS_1;
;  680          1616  2
;  681          1617  2 SS$_NORMAL
;  682          1618  2
;  683          1619  1 END;
```

```
                                                 0004 00000          .ENTRY  PSM$UPDATE_SCB, Save R2                 ; 1587
                                              5E                 0C  C2 00002          SUBL2   #12, SP
                                                                 7E  D4 00005          CLRL    CONTEXT                 ; 1591
                                                           04    AE  D4 00007          CLRL    ITEM
                                                     08    AE 020E0000 8F  D0 0000A          MOVL    #34471936, DESC         ; 1601
                                                           0C    AE  D4 00012          CLRL    DESC+4
                                              50           04    AC  D0 00015          MOVL    SCB, R0                 ; 1604
                                                         0140    C0  D4 00019          CLRL    320(R0)
                                                           08    AE  9F 0001D 1$:      PUSHAB  DESC                    ; 1606
                                                           08    AE  9F 00020          PUSHAB  ITEM
                                                           08    AE  9F 00023          PUSHAB  CONTEXT
                                                           08    AC  DD 00026          PUSHL   MESSAGE
                                        00000000G          00        04  FB 00029          CALLS   #4, SMB$READ_MESSAGE_ITEM
                                                           52        50  D0 00030          MOVL    R0, STATUS_1
                                                           1B        52  E9 00033          BLBC    STATUS_1, 3$
                                                           50    04  AC  D0 00036          MOVL    SCB, R0                 ; 1609
                                                 00      01B0    C0  04  AE  E2 0003A          BBSS    ITEM, 432(R0), 2$       ; 1610
                                                           08    AE  9F 00041 2$:      PUSHAB  DESC
                                                           08    AE  9F 00044          PUSHAB  ITEM
                                                           04    AC  9F 00047          PUSHAB  SCB
                                                     0000V       CF        03  FB 0004A          CALLS   #3, PSM$WRITE_ITEM_DX
                                                                           CC  11 0004F          BRB     1$              ; 1606
                                        00000000G          8F        52  D1 00051 3$:      CMPL    STATUS_1, #SMB$_NOMOREITEMS ; 1613
                                                                     04  13 00058          BEQL    4$
                                                           50        52  D0 0005A          MOVL    STATUS_1, R0            ; 1615
                                                                     04 0005D          RET
                                                           50        01  D0 0005E 4$:      MOVL    #1, R0                  ; 1619
                                                                     04 00061          RET

; Routine Size:  98 bytes,    Routine Base:  CODE + 03F4
```

```
;  685          1620  1 %SBTTL 'WRITE_ITEM_DX'
;  686          1621  1 !++
;  687          1622  1 ! Functional Description:
;  688          1623  1 !               NOT SUPPORTED IN V4.0.
;  689          1624  1 !
;  690          1625  1 ! Formal Parameters:
;  691          1626  1 !               ?desc
;  692          1627  1 ! Implicit Inputs:
;  693          1628  1 !               none
;  694          1629  1 !
;  695          1630  1 ! Implicit Outputs:
;  696          1631  1 !               none
;  697          1632  1 !
;  698          1633  1 ! Returned Value:
;  699          1634  1 !               none
;  700          1635  1 !
;  701          1636  1 ! Side Effects:
;  702          1637  1 !               none
;  703          1638  1 !--
;  704          1639  1 GLOBAL ROUTINE PSM$WRITE_ITEM_DX (
;  705          1640  1         SMB_CONTEXT     : REF $LONGWORD,
;  706          1641  1         ITEM_CODE       : REF $LONGWORD,
;  707          1642  1         DESCRIPTOR      : REF $BBLOCK
;  708          1643  2         ) =
;  709          1644  2 BEGIN
;  710          1645  2
;  711          1646  2 LOCAL   ITEM_ADR;
;  712          1647  2
;  713          1648  2 IF .ITEM_CODE[] GEQU SMBMSG$K_MAX_ITEM_CODE THEN RETURN PSM$_INVITMCOD;
;  714          1649  2
;  715          1650  2 ITEM_ADR = .SMB_CONTEXT[] + .PSM$MIT[.ITEM_CODE[], MIT_W_OFFSET];
;  716          1651  2
;  717          1652  2 CASE .PSM$MIT[.ITEM_CODE[], MIT_B_TYPE] FROM MIT_K_STATIC TO -1 OF
;  718          1653  2     SET
;  719          1654  2
;  720          1655  2     [MIT_K_DYNAMIC, MIT_K_STATIC]:
;  721          1656  2         COPY_DX_DX_ (.DESCRIPTOR, .ITEM_ADR);
;  722          1657  2
;  723          1658  2     [OUTRANGE]:
;  724          1659  2         CH$COPY (
;  725          1660  2             .DESC_SIZE_ (.DESCRIPTOR),          ! User item size
;  726          1661  2             .DESC_ADDR_ (.DESCRIPTOR),          ! User item address
;  727          1662  2             0,                                  ! Zero fill
;  728          1663  2             .PSM$MIT[.ITEM_CODE[], MIT_B_TYPE], ! Table item size
;  729          1664  2             .ITEM_ADR);                         ! Table item address
;  730          1665  2
;  731          1666  2     TES;
;  732          1667  2
;  733          1668  2 SS$_NORMAL
;  734          1669  1 END;


                                 003C 00000         .ENTRY  PSM$WRITE_ITEM_DX, Save R2,R3,R4,R5          ; 1638
                    51       08  BC   D0 00002       MOVL    @ITEM_CODE, R1                               ; 1647
```

```
                                39              51  D1 00006          CMPL    R1, #57                                                 ;
                                                08  1F 00009          BLSSU   1$                                                      ;
                                50 00000000G    8F  D0 0000B          MOVL    #PSM$_INVITMCOD, R0                                     ;
                                                    04 00012          RET                                                             ;
                                   00000000G0041    DF 00013 1$:      PUSHAL  PSM$MIT+2[R1]                                   ; 1649
                                52                  9E 3C 0001A       MOVZWL  @(SP)+, ITEM_ADR                                        ;
                                52          04      BC C0 0001D       ADDL2   @SMB_CONTEXT, ITEM_ADR                                  ;
                                50          0C      AC D0 00021       MOVL    DESCRIPTOR, R0                                  ; 1659
                                   00000000G0041    DF 00025          PUSHAL  PSM$MIT[R1]                                     ; 1651
                                51                  9E 98 0002C       CVTBL   @(SP)+, R1                                              ;
                    01      FE  8F                  51  8F 0002F     CASEB   R1, #-2, #1                                             ;
                                    000D                000D  00034 2$:   .WORD   3$-2$,-                                         ;
                                                                          3$-2$                                                     ;
        51          00      04  B0                  60  2C 00038      MOVC5   (R0), @4(R0), #0, R1, (ITEM_ADR)                ; 1663
                                                    62      0003E                                                                   ;
                                                    1A  11 0003F      BRB     4$                                              ; 1658
                                                    50  DD 00041 3$:  PUSHL   R0                                              ; 1655
                                                    52  DD 00043      PUSHL   ITEM_ADR                                                ;
                    00000000G   00                  02  FB 00045      CALLS   #2, STR$COPY_DX                                         ;
                                52                  50  D0 0004C      MOVL    R0, STATUS                                              ;
                                09                  52  E8 0004F      BLBS    STATUS, 4$                                              ;
                                                    52  DD 00052      PUSHL   STATUS                                                  ;
                    00000000G   00                  01  FB 00054      CALLS   #1, LIB$SIGNAL                                          ;
                                50                  01  D0 0005B 4$:  MOVL    #1, R0                                          ; 1669
                                                        04 0005E      RET                                                             ;

; Routine Size:  95 bytes,    Routine Base:  CODE + 0456
```

```
;  736          1670  1 %SBTTL 'WRITE_ITEM_R'
;  737          1671  1 ! Functional Description:
;  738          1672  1 !                       NOT SUPPORTED IN V4.0.                                                                  ;
;  739          1673  1 !
;  740          1674  1 ! Formal Parameters:
;  741          1675  1 !                       ?desc
;  742          1676  1 ! Implicit Inputs:
;  743          1677  1 !                       none
;  744          1678  1 !
;  745          1679  1 ! Implicit Outputs:
;  746          1680  1 !                       none
;  747          1681  1 !
;  748          1682  1 ! Returned Value:
;  749          1683  1 !                       none
;  750          1684  1 !
;  751          1685  1 ! Side Effects:
;  752          1686  1 !                       none
;  753          1687  1 !--
;  754          1688  1 GLOBAL ROUTINE PSM$WRITE_ITEM_R (
;  755          1689  1         SMB_CONTEXT     : REF $LONGWORD,
;  756          1690  1         ITEM_CODE       : REF $LONGWORD,
;  757          1691  1         BUFSIZ          : REF $WORD,
;  758          1692  1         BUFADR          : REF VECTOR [,BYTE]
;  759          1693  1         ) =
;  760          1694  2 BEGIN
;  761          1695  2
;  762          1696  2 LOCAL USER_DESCRIPTOR : VECTOR [2];
;  763          1697  2
;  764          1698  2 USER_DESCRIPTOR [0] = .BUFSIZ[];
;  765          1699  2 USER_DESCRIPTOR [1] = .BUFADR;
;  766          1700  2
;  767          1701  2 RETURN PSM$WRITE_ITEM_DX (.SMB_CONTEXT, .ITEM_CODE, USER_DESCRIPTOR);
;  768          1702  2
;  769          1703  1 END;


                                        0000 00000      .ENTRY  PSM$WRITE_ITEM_R, Save nothing                  ; 1688
                                5E      04   C2 00002      SUBL2   #4, SP                                          ;
                                7E      0C   BC 3C 00005  MOVZWL  @BUFSIZ, USER_DESCRIPTOR                        ; 1698
                            04  AE      10   AC D0 00009  MOVL    BUFADR, USER_DESCRIPTOR+4                       ; 1699
                                5E      DD 0000E        PUSHL   SP                                              ; 1701
                                7E      04   AC 7D 00010  MOVQ    SMB_CONTEXT, -(SP)                              ;
                            89  AF      03   FB 00014      CALLS   #3, PSM$WRITE_ITEM_DX                           ;
                                        04 00018        RET                                                     ; 1703

; Routine Size:  25 bytes,    Routine Base:  CODE + 04B5
```

```
;  771          1704  1 %SBTTL 'ALLOCATE_MEMORY'
;  772          1705  1 ! Functional Description:
;  773          1706  1 !                   Allocates memory through RTL routines.
;  774          1707  1 !
;  775          1708  1 ! Formal Parameters:
;  776          1709  1 !                   BYTES   - number of bytes of memory to allocate
;  777          1710  1 !
;  778          1711  1 ! Implicit Inputs:
;  779          1712  1 !                   none
;  780          1713  1 !
;  781          1714  1 ! Implicit Outputs:
;  782          1715  1 !                   none
;  783          1716  1 !
;  784          1717  1 ! Returned Value:
;  785          1718  1 !                   address of allocated memory
;  786          1719  1 !
;  787          1720  1 ! Side Effects:
;  788          1721  1 !                   none
;  789          1722  1 !--
;  790          1723  1 ROUTINE ALLOCATE_MEMORY (
;  791          1724  1         BYTES
;  792          1725  1         )       : =
;  793          1726  2 BEGIN
;  794          1727  2
;  795          1728  2 LOCAL
;  796          1729  2         MEMORY
;  797          1730  2         ;
;  798          1731  2
;  799          1732  2 SIGNAL_IF_ERROR_ (LIB$GET_VM (BYTES, MEMORY));
;  800          1733  2
;  801          1734  2 RETURN .MEMORY;
;  802          1735  2
;  803          1736  1 END;


                                0004 00000 ALLOCATE_MEMORY:
                                           .WORD   Save R2                         ;  1723
                        5E              04  C2 00002        SUBL2   #4, SP          ;
                                        5E  DD 00005        PUSHL   SP              ;  1732
                                04      AC  9F 00007        PUSHAB  BYTES           ;
              00000000G         00      02  FB 0000A        CALLS   #2, LIB$GET_VM  ;
                        52              50  D0 00011        MOVL    R0, STATUS      ;
                        09              52  E8 00014        BLBS    STATUS, 1$      ;
                                        52  DD 00017        PUSHL   STATUS          ;
              00000000G         00      01  FB 00019        CALLS   #1, LIB$SIGNAL  ;
                        50              6E  D0 00020 1$:     MOVL    MEMORY, R0      ;  1734
                                            04 00023        RET                     ;  1736

; Routine Size:  36 bytes,    Routine Base:  CODE + 04CE
```

```
;  805          1737  1 END
;  806          1738  0 ELUDOM


                                                        .EXTRN  LIB$SIGNAL

;                                  PSECT SUMMARY

;       Name                    Bytes                           Attributes

;  DATA                           16  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  CODE                         1266  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                                Library Statistics

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

;  _$255$DUA28:[SYSLIB]LIB.L32;1          18619        23         0      1000          00:01.9


;                                     COMMAND QUALIFIERS

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

; Size:         1263 code + 19 data bytes
; Run Time:        00:31.3
; Elapsed Time:    01:06.7
; Lines/CPU Min:    3332
; Lexemes/CPU-Min: 28905
; Memory Used:  212 pages
; Compilation Complete
```

MESSAGE
LIS

OUTPUT
LIS

SEPARATE
LIS

INPUT
LIS

PRTSMB
LIS

MEMORY
LIS

PSMMSG
LIS