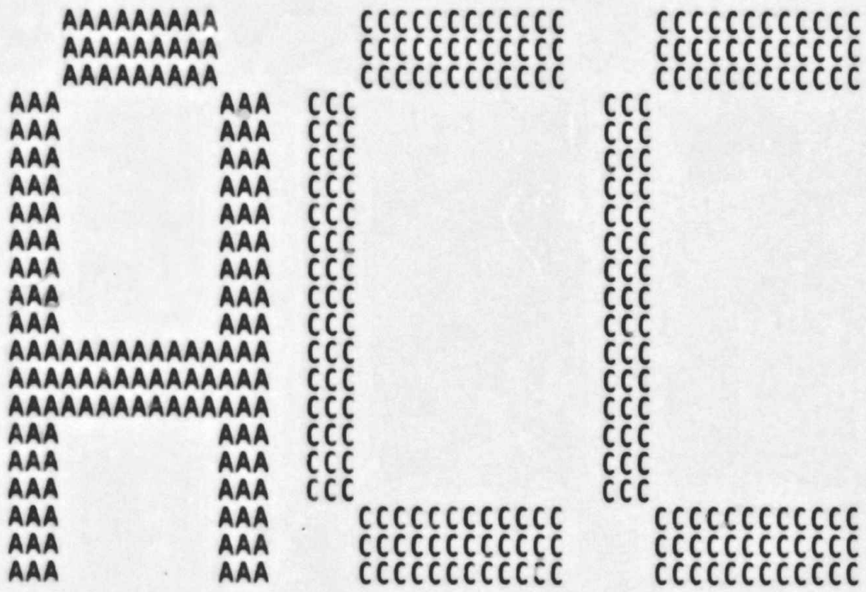
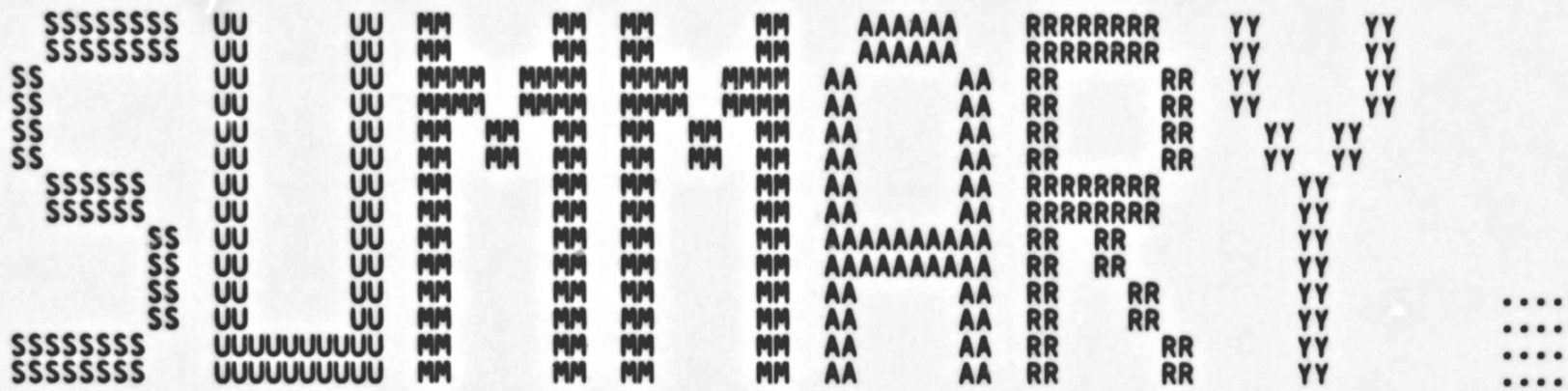
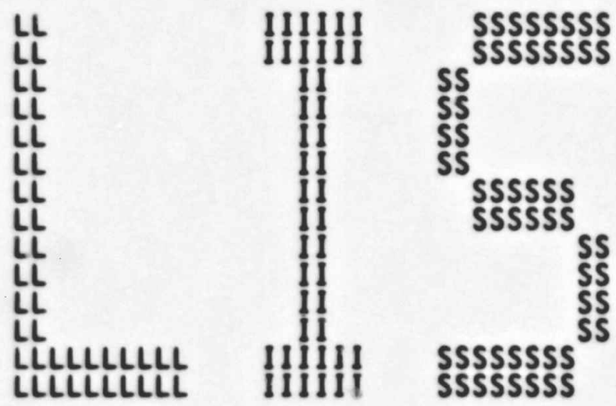
TITLE1**AH-BI13A-BE
TITLE2**VAX/VMS V4.0
FICHENUMBER**1



FILEID**SUMMARY





```
;  1          0001  0 MODULE
;  2          0002  0 SUMMARY (IDENT = 'V04-000') =
;  3          0003  1 BEGIN
;  4          0004  1
;  5          0005  1 !
;  6          0006  1 !*****************************************************************************
;  7          0007  1 !*                                                                           *
;  8          0008  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
;  9          0009  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
; 10          0010  1 !*  ALL RIGHTS RESERVED.                                                    *
; 11          0011  1 !*                                                                         *
; 12          0012  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
; 13          0013  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
; 14          0014  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER  *
; 15          0015  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
; 16          0016  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
; 17          0017  1 !*  TRANSFERRED.                                                            *
; 18          0018  1 !*                                                                         *
; 19          0019  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
; 20          0020  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
; 21          0021  1 !*  CORPORATION.                                                            *
; 22          0022  1 !*                                                                         *
; 23          0023  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
; 24          0024  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
; 25          0025  1 !*                                                                         *
; 26          0026  1 !*                                                                         *
; 27          0027  1 !*****************************************************************************
; 28          0028  1
; 29          0029  1 !++
; 30          0030  1 ! FACILITY:  ACC, Account file dumper
; 31          0031  1 !
; 32          0032  1 ! ABSTRACT:
; 33          0033  1 !
; 34          0034  1 !       This module contains routines used to build and manipulate
; 35          0035  1 !       sort and summarization keys as well as routines to accumulate
; 36          0036  1 !       and report summarization totals and general totals.
; 37          0037  1 !
; 38          0038  1 ! ENVIRONMENT:
; 39          0039  1 !
; 40          0040  1 !       VAX/VMS operating system. unprivileged user mode,
; 41          0041  1 !
; 42          0042  1 ! AUTHOR:  Greg Robert and Steve Forgey, January 1982
; 43          0043  1 !
; 44          0044  1 ! Modified by:
; 45          0045  1 !
; 46          0046  1 !       V03-005 DAS0002         David Solomon   12-Jan-1984
; 47          0047  1 !               Get ACCDEF.REQ from SRC$, not MSRC$.
; 48          0048  1 !
; 49          0049  1 !       V03-004 DAS0001         David Solomon   03-Jan-1984
; 50          0050  1 !               Change references to messages from ACC_ to ACC$_. Use new
; 51          0051  1 !               literal for summation symbol table number. In RELEASE_TO_SORT,
; 52          0052  1 !               get key table index number from new table, SORT_INDEX_TABLE, not
; 53          0053  1 !               from key start position word in key description in SORT_TABLE.
; 54          0054  1 !
; 55          0055  1 !       V03-003 SPF0082         Steve Forgey    Feb-11-1982
; 56          0056  1 !               Don't write records rejected by SORT if the /REJECTED
; 57          0057  1 !               qualifier was not present.
```

```
1 !
1 !           V03-002 SPF0081         Steve Forgey    Feb-06-1982
1 !                   Output delta times in "!20L !%T" format.
1 !
1 !           V03-001 SPF0071         Steve Forgey    Jan-23-1982
1 !                   Include report item name in bar chart reports.
1 !
1 !--
1
1 !-----------------------------------------------------------------------------+
1 !                                                                             !
1 !                               INCLUDE  FILES                                !
1 !                                                                             !
1 !-----------------------------------------------------------------------------+
1
1 REQUIRE 'SRC$:ACCDEF';          ! Common ACC definitions
```

```
;    75          0912  1 !----------------------------------------------------------------------+
;    76          0913  1 !                                                                      |
;    77          0914  1 !                          TABLE  OF  CONTENTS                         |
;    78          0915  1 !                                                                      |
;    79          0916  1 !----------------------------------------------------------------------+
;    80          0917  1
;    81          0918  1 UNDECLARE
;    82          0919  1         RELEASE_TO_SORT,
;    83          0920  1         SUMMARIZE,
;    84          0921  1         WRITE_SUMMARY,
;    85          0922  1         FIND_WATERMARK,
;    86          0923  1         WRITE_BAR_GRAPH,
;    87          0924  1         WRITE_TOTALS;
;    88          0925  1
;    89          0926  1 FORWARD ROUTINE
;    90          0927  1         RELEASE_TO_SORT,                ! Build keys and release to sort
;    91          0928  1         SUMMARIZE,                      ! Summation main control loop
;    92          0929  1         BUILD_SUMMARY,                  ! Builds a summarization key
;    93          0930  1         FIND_SUMMARY,                   ! Locates and allocates summary buckets
;    94          0931  1         ACCUMULATE_SUMMARY,             ! Accumulates summary totals
;    95          0932  1         WRITE_SUMMARY,                  ! Output summary at end of file
;    96          0933  1         FIND_WATERMARK,                 ! Determine peak values
;    97          0934  1         WRITE_BAR_GRAPH,                ! Output bar graphs
;    98          0935  1         SETUP_GRAPH,                    ! Prepare to do bar graph
;    99          0936  1         WRITE_TOTALS;                   ! Output totals at end of program
;   100          0937  1
```

```
;  102          0938  1 !----------------------------------------------------------------------+
;  103          0939  1 !                                                                      |
;  104          0940  1 !                          GENERAL  STORAGE  DEFINITIONS               |
;  105          0941  1 !                                                                      |
;  106          0942  1 !----------------------------------------------------------------------+
;  107          0943  1
;  108          0944  1 OWN
;  109          0945  1         hwm_bucket: ref vector [],      ! Pointer to high watermark bucket
;  110          0946  1         lwm_bucket: ref vector [],      ! Pointer to low watermark bucket
;  111          0947  1         key_desc: bblock [dsc$k_d_bln]  ! Dynamic key symbol descriptor
;  112          0948  1                 preset([dsc$b_class] = dsc$k_class_d),
;  113          0949  1         bar_fao: bblock [dsc$k_d_bln]   ! Dynamic work descriptor
;  114          0950  1               preset([dsc$b_class] = dsc$k_class_d),
;  115          0951  1         bar_char: bblock [dsc$k_d_bln]  ! Dynamic work descriptor
;  116          0952  1               preset([dsc$b_class] = dsc$k_class_d),
;  117          0953  1         desc: bblock [dsc$k_d_bln]      ! Dynamic work descriptor
;  118          0954  1               preset([dsc$b_class] = dsc$k_class_d),
;  119          0955  1         wdesc: bblock [dsc$k_d_bln]     ! Dynamic work descriptor
;  120          0956  1               preset([dsc$b_class] = dsc$k_class_d);
;  121          0957  1
;  122          0958  1
```

```
;  124     0959  1 GLOBAL ROUTINE RELEASE_TO_SORT (INPUT, KEYS, TABLE, OUTPUT) =
;  125     0960  1
;  126     0961  1 !----
;  127     0962  1 !
;  128     0963  1 ! Functional description
;  129     0964  1 !
;  130     0965  1 !       This routine builds a concatenated key string using standard
;  131     0966  1 !       sort key descriptors. And then release the record to the sort
;  132     0967  1 !       package.
;  133     0968  1 !
;  134     0969  1 ! Input parameters
;  135     0970  1 !
;  136     0971  1 !       INPUT   = Address of the input record
;  137     0972  1 !       KEYS    = Address of a key descriptor list head
;  138     0973  1 !       TABLE   = Address of key table
;  139     0974  1 !
;  140     0975  1 ! Output parameters
;  141     0976  1 !
;  142     0977  1 !       OUTPUT  = Address of a buffer to recieve the concatenated key
;  143     0978  1 !       Any errors encountered are RETURNed immediately.
;  144     0979  1 !       TRUE is returned on a normal exit.
;  145     0980  1 !
;  146     0981  1 !----
;  147     0982  1
;  148     0983  2 BEGIN
;  149     0984  2
;  150     0985  2 EXTERNAL
;  151     0986  2         selected,                       ! Selected record count
;  152     0987  2         rejected_rab,                   ! Rejected record output file RAB
;  153     0988  2         version,                        ! Record format version
;  154     0989  2         sort_index_table: vector [max_sort+1, byte];
;  155     0990  2                                         ! Sort key index table.
;  156     0991  2
;  157     0992  2 MAP
;  158     0993  2     input: ref bblock,
;  159     0994  2     keys: ref blockvector [,quad,byte],  ! Key descriptors
;  160     0995  2     table: ref blockvector [,3];         ! Key table
;  161     0996  2
;  162     0997  2 LOCAL
;  163     0998  2         index,                          ! Key table index
;  164     0999  2         srcptr,                         ! Source string pointer
;  165     1000  2         srclen,                         ! Source string length
;  166     1001  2         desc:   vector [2, long];       ! Temporary string descriptor
;  167     1002  2
;  168     1003  2 !
;  169     1004  2 !INITIALIZE THE DESCRIPTOR --
;  170     1005  2 !       Initialize a descriptor with the address and size of the
;  171     1006  2 !       sort record including keys.
;  172     1007  2 !
;  173     1008  2
;  174     1009  2 SELECTONEU .version of
;  175     1010  2         SET
;  176     1011  2         [acr$k_version2]:
;  177     1012  2                 desc [0] = .input + .input [acc$w_msgsiz] - .output;
;  178     1013  2         [acr$k_version3t, acr$k_version3]:
;  179     1014  2                 desc [0] = .input + .input [acr$w_length] - .output;
;  180     1015  2         TES;
```

```
;  181          1016  2
;  182          1017  2 desc [1] = .output;
;  183          1018  2
;  184          1019  2 Incr i to .(.keys-2)<0,8> do
;  185          1020  3         BEGIN
;  186          1021  3         index = .sort_index_table [.i];          ! Key table index
;  187          1022  3         srclen = .(.table [.index, sort_desc]); ! Source string length
;  188          1023  3         if      .srclen eql 0
;  189          1024  4         then    BEGIN
;  190          1025  4                 selected = .selected - 1;
;  191          1026  5                 if      PRESENT (REJECTED)
;  192          1027  4                 then    perform (write_binary (.input, rejected_rab));
;  193          1028  4                 return true;
;  194          1029  3                 END;
;  195          1030  3         srcptr = .(.table [.index, sort_desc] + 4);! Source string address
;  196          1031  3         output = ch$copy ( .srclen, .srcptr,
;  197          1032  3                 0,                                ! Fill character
;  198          1033  3                 .keys [.i, key_w_length],         ! Destination length
;  199          1034  3                 .output);                         ! Destination address
;  200          1035  2         END;
;  201          1036  2
;  202          1037  2 !
;  203          1038  2 ! RELEASE TO SORT --
;  204          1039  2 !       Release the record to the sort package.
;  205          1040  2 !
;  206          1041  2
;  207          1042  2 Return sor$release_rec (desc);
;  208          1043  2
;  209          1044  1 END;
```

```
                        .TITLE  SUMMARY
                        .IDENT  \V04-000\

                        .PSECT  DATA,NOEXE,2

                00000 HWM_BUCKET:
                        .BLKB   4
                00004 LWM_BUCKET:
                        .BLKB   4
00#             00008 KEY_DESC:
                        .BYTE   0[3]                                    ;
02              0000B           .BYTE   2                               ;
                0000C           .BLKB   4
00#             00010 BAR_FAO:  .BYTE   0[3]                            ;
02              00013           .BYTE   2                               ;
                00014           .BLKB   4
00#             00018 BAR_CHAR:
                        .BYTE   0[3]                                    ;
02              0001B           .BYTE   2                               ;
                0001C           .BLKB   4
00#             00020 DESC:     .BYTE   0[3]                            ;
02              00023           .BYTE   2                               ;
                00024           .BLKB   4
00#             00028 WDESC:    .BYTE   0[3]                            ;
02              0002B           .BYTE   2                               ;
                0002C           .BLKB   4
```

```
                                        .EXTRN  ACC$_INVACCREC, ACC$_TOTAL
                                        .EXTRN  ACC$_MERGE, ACC$_INPUT
                                        .EXTRN  ACC$_TITLETRUNC
                                        .EXTRN  ADD_SYMBOL, ALLOCATE
                                        .EXTRN  FIND_WATERMARK, HANDLER
                                        .EXTRN  LIB$ADDX, LIB$CVT_DTB
                                        .EXTRN  LIB$CVT_HTB, LIB$CVT_TIME
                                        .EXTRN  LIB$DAY, LIB$FILE_SCAN
                                        .EXTRN  LIB$ICHAR, LIB$LOOKUP_KEY
                                        .EXTRN  LIB$SUBX, LIB$SYS_ASCTIM
                                        .EXTRN  LIB$SYS_FAO, LIB$SYS_FAOL
                                        .EXTRN  LIB$TPARSE, LOG_FILENAME
                                        .EXTRN  LOOKUP_SYMBOL, MAP_QUALIFIERS
                                        .EXTRN  PARSE_OUTPUT_FILES
                                        .EXTRN  RELEASE_TO_SORT
                                        .EXTRN  SCAN_SYMBOLS, SHOW_RECORD
                                        .EXTRN  SOR$END_SORT, SOR$INIT_SORT
                                        .EXTRN  SOR$RELEASE_REC
                                        .EXTRN  SOR$RETURN_REC, SOR$SORT_MERGE
                                        .EXTRN  STR$APPEND, STR$COMPARE
                                        .EXTRN  STR$DUPL_CHAR, STR$LEFT
                                        .EXTRN  STR$PREFIX, STR$REPLACE
                                        .EXTRN  STR$RIGHT, STRIP_NEGATOR
                                        .EXTRN  STRIP_TRAIL, SUMMARIZE
                                        .EXTRN  SYS$NUMTIM, TRANSLATE_STATUS
                                        .EXTRN  WRITE_BAR_GRAPH
                                        .EXTRN  WRITE_BINARY, WRITE_SUMMARY
                                        .EXTRN  WRITE_TOTALS, SELECTED
                                        .EXTRN  REJECTED_RAB, VERSION
                                        .EXTRN  SORT_INDEX_TABLE
                                        .EXTRN  QUALIFIERS

                                        .PSECT  CODE,NOWRT,2

                        0FFC 00000      .ENTRY  RELEASE_TO_SORT, Save R2,R3,R4,R5,R6,R7,R8,-; 0959
                                                R9,R10,R11                              ;
                5E                08  C2 00002       SUBL2   #8, SP                  ; 1009
                50 00000000G      00  D0 00005       MOVL    VERSION, R0             ; 1011
                                  05  13 0000C       BEQL    1$                      ; 1013
                02                    50  D1 0000E   CMPL    R0, #2                  ;
                                      10  1A 00011   BGTRU   2$                      ; 1014
                51            04  AC  D0 00013 1$:   MOVL    INPUT, R1               ;
                50            02  A1  3C 00017       MOVZWL  2(R1), R0               ;
                50                51  C0 0001B       ADDL2   R1, R0                  ;
        6E      50            10  AC  C3 0001E       SUBL3   OUTPUT, R0, DESC        ;
              04 AE           10  AC  D0 00023 2$:    MOVL    OUTPUT, DESC+4          ; 1017
                58            08  AC  D0 00028       MOVL    KEYS, R8                ; 1019
                5A            FE  A8  9A 0002C       MOVZBL  -2(R8), R10             ;
                57                01  CE 00030       MNEGL   #1, I                   ; 1022
                                  51  11 00033       BRB     6$                      ;
                56 00000000G00047 9A 00035 3$:        MOVZBL  SORT_INDEX_TABLE[I], INDEX ; 1021
        50      56                0C  C5 0003D       MULL3   #12, INDEX, R0          ; 1022
                50            0C  AC  C0 00041       ADDL2   TABLE, R0               ;
                52            04  A0  D0 00045       MOVL    4(R0), R2               ;
                59                62  D0 00049       MOVL    (R2), SRCLEN            ;
                                  25  12 0004C       BNEQ    5$                      ; 1023
```

```
                          00000000G  00    D7 0004E          DECL    SELECTED                                        ; 1025
                 13 00000000G  00    02    E1 00054          BBC     #2, QUALIFIERS+2, 4$                            ; 1026
                          00000000G  00    9F 0005C          PUSHAB  REJECTED_RAB                                    ; 1027
                                04    AC    DD 00062          PUSHL   INPUT                                           ;
                    00000000G  00       02    FB 00065      CALLS   #2, WRITE_BINARY                                ;
                               24       50    E9 0006C      BLBC    STATUS, 7$                                      ;
                               50       01    D0 0006F 4$:  MOVL    #1, R0                                          ; 1028
                                                04 00072      RET                                                     ;
                               5B    04    A2    D0 00073 5$:  MOVL    4(R2), SRCPTR                                   ; 1030
                                     06 A847    7F 00077      PUSHAQ  6(R8)[I]                                        ; 1034
           9E        00          6B          59    2C 0007B      MOVC5   SRCLEN, (SRCPTR), #0, @(SP)+, @OUTPUT       ;
                                10    BC          00080                                                             ;
                       10       AC          53    D0 00082      MOVL    R3, OUTPUT                                      ;
                      AB        57          5A    F3 00086 6$:  AOBLEQ  R10, I, 3$                                  ; 1019
                                            5E    DD 0008A      PUSHL   SP                                          ; 1042
                    00000000G  00          01    FB 0008C      CALLS   #1, SOR$RELEASE_REC                         ;
                                                04 00093 7$:  RET                                                     ; 1044

; Routine Size:  148 bytes,    Routine Base:  CODE + 0000
```

```
;   211          1045  1 GLOBAL ROUTINE SUMMARIZE (BUFFER) =
;   212          1046  1
;   213          1047  1 !----
;   214          1048  1 !
;   215          1049  1 ! Functional description
;   216          1050  1 !
;   217          1051  1 !       This routine is part of summarization logic.  It is the main
;   218          1052  1 !       control routine.  It is entered when a record is selected and
;   219          1053  1 !       the user has specified /SUMMARY.
;   220          1054  1 !
;   221          1055  1 !       A symbol is constructed from the record which uniquely identifies
;   222          1056  1 !       the summarization 'bucket' to which this record belongs.  The
;   223          1057  1 !       construction is driven by the list of summarization keys supplied
;   224          1058  1 !       by the user.
;   225          1059  1 !
;   226          1060  1 !       Example:  If the user specified:
;   227          1061  1 !
;   228          1062  1 !                       ACC /SUMMARY=(DAY, USER, ACCOUNT)
;   229          1063  1 !
;   230          1064  1 !       then a symbol (key) would be build out of a date string, username,
;   231          1065  1 !       and account name from the record:
;   232          1066  1 !
;   233          1067  1 !                       YYYY MM DD USER---- ACCOUNT-----
;   234          1068  1 !                       1980 10 21 ROBERT   VMS
;   235          1069  1 !
;   236          1070  1 !       Next the summarization bucket for the key is located.  If not found
;   237          1071  1 !       then a new bucket is allocated and intialized.
;   238          1072  1 !
;   239          1073  1 !       Finally the bucket is updated with the contents of the record.  Only
;   240          1074  1 !       those fields requested by the user via the /REPORT qualifier are
;   241          1075  1 !       summed.  Note that some fields are summed, some are mazimized, and
;   242          1076  1 !       some are averaged.
;   243          1077  1 !
;   244          1078  1 ! Input parameters
;   245          1079  1 !
;   246          1080  1 !       BUFFER  =       Address of a record to be summed
;   247          1081  1 !
;   248          1082  1 ! Output parameters
;   249          1083  1 !
;   250          1084  1 !       Any errors encountered are RETURNed immediately.
;   251          1085  1 !
;   252          1086  1 !----
;   253          1087  1
;   254          1088  2 BEGIN
;   255          1089  2
;   256          1090  2 LOCAL
;   257          1091  2         bucket;                         ! Receives address of summation bucket
;   258          1092  2
;   259          1093  2 Perform (build_summary (.buffer, key_desc));    ! Build a summarization key
;   260          1094  2
;   261          1095  2 Perform (find_summary (key_desc, bucket));      ! Locate the summary bucket
;   262          1096  2
;   263          1097  2 Perform (accumulate_summary (.buffer, .bucket));! Merge this record
;   264          1098  2
;   265          1099  2
;   266          1100  2 return true;
;   267          1101  1 END;
```

```
                                        0004 00000          .ENTRY  SUMMARIZE, Save R2                                      ; 1045
                52 00000000'    EF   9E 00002          MOVAB   KEY_DESC, R2                                                   ;
                5E              04   C2 00009          SUBL2   #4, SP                                                         ;
                                52   DD 0000C          PUSHL   R2                                                             ; 1093
                          04    AC   DD 0000E          PUSHL   BUFFER                                                         ;
          0000V CF              02   FB 00011          CALLS   #2, BUILD_SUMMARY                                              ;
                1C              50   E9 00016          BLBC    STATUS, 1$                                                     ;
                            4004 8F   BB 00019          PUSHR   #^M<R2,SP>                                                     ; 1095
          0000V CF              02   FB 0001D          CALLS   #2, FIND_SUMMARY                                               ;
                10              50   E9 00022          BLBC    STATUS, T$                                                     ;
                                6E   DD 00025          PUSHL   BUCKET                                                         ; 1097
                          04    AC   DD 00027          PUSHL   BUFFER                                                         ;
          0000V CF              02   FB 0002A          CALLS   #2, ACCUMULATE_SUMMARY                                         ;
                03              50   E9 0002F          BLBC    STATUS, 1$                                                     ;
                50              01   D0 00032          MOVL    #1, R0                                                         ; 1100
                                     04 00035 1$:       RET                                                                    ; 1101

; Routine Size:  54 bytes,    Routine Base:  CODE + 0094
```

```
;  269          1102  1 ROUTINE BUILD_SUMMARY (BUFFER, KEY) =
;  270          1103  1
;  271          1104  1 !----
;  272          1105  1 !
;  273          1106  1 ! Functional description
;  274          1107  1 !
;  275          1108  1 !       This routine is part of summarization logic.  A summation key is
;  276          1109  1 !       built for the record contained in BUFFER.  The user has supplied
;  277          1110  1 !       a list of summation keys via the /SUMMARY=(key1, key2, ...) syntax.
;  278          1111  1 !       A string is built by appending the values associated with each key
;  279          1112  1 !       for this particular record.
;  280          1113  1 !
;  281          1114  1 ! Input parameters
;  282          1115  1 !
;  283          1116  1 !       BUFFER  =       Address of a record
;  284          1117  1 !       KEY     =       Address of a dynamic descriptor to receive symbol
;  285          1118  1 !
;  286          1119  1 ! Output parameters
;  287          1120  1 !
;  288          1121  1 !       A summary key is built in the output key buffer.
;  289          1122  1 !       Any errors encountered are RETURNed immediately.
;  290          1123  1 !
;  291          1124  1 !----
;  292          1125  1
;  293          1126  2 BEGIN
;  294          1127  2
;  295          1128  2 EXTERNAL
;  296          1129  2         sum_key_fao,            ! FAO describing key
;  297          1130  2         sum_key_value;          ! Values matching FAO
;  298          1131  2 !
;  299          1132  2 ! SUM_KEY_FAO is a descriptor of the desired key value.  It is built out
;  300          1133  2 ! of the individual descriptors associated with each possible summation key.
;  301          1134  2 !
;  302          1135  2 ! SUM_KEY_VALUE is a table of addresses of values suitable for a FAOL call.
;  303          1136  2 ! This table matches the set of FAO directives in SUM_KEY_FAO and is built
;  304          1137  2 ! in a similar way.
;  305          1138  2 !
;  306          1139  2 ! For construction details see PARSE_KEYS.
;  307          1140  2 !
;  308          1141  2
;  309          1142  2 perform (lib$sys_faol (sum_key_fao, 0, .key, sum_key_value));
;  310          1143  2
;  311          1144  2 return true;
;  312          1145  1 END;


                                                        .EXTRN  SUM_KEY_FAO, SUM_KEY_VALUE

                                        0000 00000 BUILD_SUMMARY:
                                                        .WORD   Save nothing                    ; 1102
                              00000000G   00  9F 00002          PUSHAB  SUM_KEY_VALUE         ; 1142
                                    08    AC  DD 00008          PUSHL   KEY                   ;
                                          7E  D4 0000B          CLRL    -(SP)                 ;
                              00000000G   00  9F 0000D          PUSHAB  SUM_KEY_FAO           ;
                    00000000G   00              04  FB 00013    CALLS   #4, LIB$SYS_FAOL      ;
                                03              50  E9 0001A    BLBC    STATUS, 1$            ;
```

```
                                50              01  D0 0001D                   MOVL    #1, R0                                  ; 1144
                                                04 00020 1$:                   RET                                             ; 1145

; Routine Size:  33 bytes,    Routine Base:  CODE + 00CA
```

```
;  314          1146  1 ROUTINE FIND_SUMMARY (KEY, BUCKET_ADR) =
;  315          1147  1
;  316          1148  1 !----
;  317          1149  1 !
;  318          1150  1 ! Functional description
;  319          1151  1 !
;  320          1152  1 !       This routine is part of summarization logic.  Given a summation
;  321          1153  1 !       key, this routine locates the matching summation bucket.  If no
;  322          1154  1 !       match is found, then a new bucket is allocated and a entry is
;  323          1155  1 !       made in the symbol table for this key.  The value associated with
;  324          1156  1 !       the symbol is the address of the summation bucket.
;  325          1157  1 !
;  326          1158  1 ! Input parameters
;  327          1159  1 !
;  328          1160  1 !       KEY             =       Address of descriptor or summation key
;  329          1161  1 !       BUCKET_ADR      =       Address of longword to recieve bucket address
;  330          1162  1 !
;  331          1163  1 ! Output parameters
;  332          1164  1 !
;  333          1165  1 !       BUCKET_ADR is loaded with the address of the summation bucket.
;  334          1166  1 !       Any errors encountered are RETURNed immediately.
;  335          1167  1 !
;  336          1168  1 !----
;  337          1169  1
;  338          1170  2 BEGIN
;  339          1171  2
;  340          1172  2 EXTERNAL
;  341          1173  2         bucket_size;                    ! Bucket size to allocate in bytes
;  342          1174  2
;  343          1175  2
;  344          1176  2 If not lookup_symbol (summation_table, .key, .bucket_adr)       ! New symbol?
;  345          1177  3 then    BEGIN
;  346          1178  3         perform (allocate (.bucket_size, .bucket_adr));         ! Y, get bucket
;  347          1179  3         ch$fill (0, .bucket_size, ..bucket_adr);                ! Zero it
;  348        P 1180  3         perform (add_symbol
;  349          1181  3                 (summation_table, .key, ..bucket_adr));         ! Enter symbol
;  350          1182  2         END;
;  351          1183  2
;  352          1184  2 return true;
;  353          1185  2
;  354          1186  1 END;


                                                        .EXTRN  BUCKET_SIZE

                                00FC 00000 FIND_SUMMARY:
                                                        .WORD   Save R2,R3,R4,R5,R6,R7          ; 1146
                        57 00000000G    00  9E 00002           MOVAB   BUCKET_SIZE, R7         ;
                        7E          04  AC  7D 00009           MOVQ    KEY, -(SP)              ; 1176
                                        3F  DD 0000D           PUSHL   #63                     ;
                00000000G   00          03  FB 0000F           CALLS   #3, LOOKUP_SYMBOL       ;
                            2A          50  E8 00016           BLBS    R0, 1$                  ;
                                    08  AC  DD 00019           PUSHL   BUCKET_ADR              ; 1178
                                        67  DD 0001C           PUSHL   BUCKET_SIZE             ;
                00000000G   00          02  FB 0001E           CALLS   #2, ALLOCATE            ;
                            1E          50  E9 00025           BLBC    STATUS, 2$              ;
```

```
                          56      08  BC  D0 00028          MOVL    @BUCKET_ADR, R6                                 ; 1179
          67              00      6E      00  2C 0002C      MOVC5   #0, (SP), #0, BUCKET_SIZE, (R6)                 ;
                                          66      00031                                                           ;
                                          56  DD  00032      PUSHL   R6                                          ; 1181
                                      04  AC  DD  00034      PUSHL   KEY                                         ;
                                          3F  DD  00037      PUSHL   #63                                         ;
                      00000000G   00          03  FB  00039      CALLS   #3, ADD_SYMBOL                          ;
                                  03          50  E9  00040      BLBC    STATUS, 2$                              ;
                                  50          01  D0  00043 1$:  MOVL    #1, R0                                  ; 1184
                                              04  00046 2$:  RET                                                  ; 1186

; Routine Size:  71 bytes,    Routine Base:  CODE + 00EB
```

```
;   356          1187  1 ROUTINE ACCUMULATE_SUMMARY (BUFFER, BUCKET) =
;   357          1188  1
;   358          1189  1 !----
;   359          1190  1 !
;   360          1191  1 ! Functional description
;   361          1192  1 !
;   362          1193  1 !       This routine is part of summarization logic.  It is called to
;   363          1194  1 !       merge a given record into its summation bucket.  The values to
;   364          1195  1 !       be merged are determined by the list of values to be reported
;   365          1196  1 !       given with the /REPORT=(val1, val2, ...) syntax, plus some
;   366          1197  1 !       standard values that are always summarized.
;   367          1198  1 !
;   368          1199  1 !       Summation bucket elements may be manipulated in any of the
;   369          1200  1 !       following ways:
;   370          1201  1 !
;   371          1202  1 !               Addition        (e.g. connect time)
;   372          1203  1 !               Integration     (e.g. CPUTIM * WSPEAK)
;   373          1204  1 !               Incrementing    (e.g. counting records)
;   374          1205  1 !               Maximizing      (e.g. generating a watermark)
;   375          1206  1 !
;   376          1207  1 ! Input parameters
;   377          1208  1 !
;   378          1209  1 !       BUFFER  =       Address of a record to be summed
;   379          1210  1 !       BUCKET  =       Address of a summation bucket
;   380          1211  1 !
;   381          1212  1 ! Output parameters
;   382          1213  1 !
;   383          1214  1 !       Any errors encountered are RETURNed immediately.
;   384          1215  1 !
;   385          1216  1 !----
;   386          1217  1
;   387          1218  2 BEGIN
;   388          1219  2
;   389          1220  2 EXTERNAL
;   390          1221  2         report_items,                   ! Number of report items
;   391          1222  2         report_value: vector [];        ! Table of pointers to offsets
;   392          1223  2
;   393          1224  2 MAP
;   394          1225  2         bucket: ref vector [],          ! Summary array
;   395          1226  2         buffer: ref bblock [];          ! Record buffer
;   396          1227  2
;   397          1228  2 LOCAL
;   398          1229  2         index;                          ! Index to summation bucket item
;   399          1230  2
;   400          1231  2 !
;   401          1232  2 ! Using the report_value table build by parse_keys accumulate data
;   402          1233  2 ! in the summary array.  Each entry in the report_value vector contains
;   403          1234  2 ! the address of a structure that describes a report item.
;   404          1235  2 !
;   405          1236  2
;   406          1237  2 Index = 1;                              ! First bucket_item reserved
;   407          1238  2
;   408          1239  2 Incr i to .report_items - 1 do
;   409          1240  3   BEGIN
;   410          1241  3   BIND rep_item = .report_value [.i]: vector [];
;   411          1242  3   case .rep_item [sum_ent_type] from 0 to 5 of
;   412          1243  3     SET
```

```
   3          [sum_type_add]:
   3              bucket [.index]  = .bucket [.index] + ..rep_item [sum_ent_adr];
   3
   3          [sum_type_addx]:
P  3              perform (lib$addx (
P  3                      bucket [.index], .rep_item [sum_ent_adr],
   3                      bucket [.index], rep_item [sum_ent_bsize]));
   3
   3          [sum_type_peak]:
   3              bucket [.index] = maxu (.bucket [.index], ..rep_item [sum_ent_adr]);
   3
   3          [sum_type_incr]:
   3              bucket [.index] = .bucket [.index] + 1;
   3
   3          [sum_type_type]:
   3              if .buffer [acc$w_msgtyp] eqlu .rep_item [sum_ent_adr]
   3              then bucket [.index] = .bucket [.index] + 1;
   3
   3          [sum_type_elap]:
   3              0;
   3
   3          TES;
   3      index = .index + .rep_item [sum_ent_bsize];
   2      END;
   2
   2  return true;
   1  END;
```

```
                                                                        .EXTRN  REPORT_ITEMS, REPORT_VALUE

                                                     003C 00000 ACCUMULATE_SUMMARY:
                                                                        .WORD   Save R2,R3,R4,R5                  ; 1187
                                  52           01  D0 00002             MOVL    #1, INDEX                         ; 1237
                                  55 00000000G 00  D0 00005             MOVL    REPORT_ITEMS, R5                  ; 1239
                                  54           01  CE 0000C             MNEGL   #1, I                             ;
                                               60  11 0000F             BRB     10$                               ;
                                  53 00000000G0044 D0 00011 1$:         MOVL    REPORT_VALUE[I], R3               ; 1241
                       05         00           63  CF 00019             CASEL   (R3), #0, #5                      ; 1242
   004C                002D     0014         000C     0001D 2$:         .WORD   3$-2$,-                           ;
                                0050         0043     00025                     4$-2$,-                           ;
                                                                                5$-2$,-                           ;
                                                                                8$-2$,-                           ;
                                                                                7$-2$,-                           ;
                                                                                9$-2$                             ;
                          08 BC42        04    B3  C0 00029 3$:         ADDL2   @4(R3), @BUCKET[INDEX]            ; 1245
                                               3C  11 0002F             BRB     9$                                ;
                                         08    A3  9F 00031 4$:         PUSHAB  8(R3)                             ; 1250
                                         08  BC42  DF 00034             PUSHAL  @BUCKET[INDEX]                    ;
                                         04    A3  DD 00038             PUSHL   4(R3)                             ;
                                         08  BC42  DF 0003B             PUSHAL  @BUCKET[INDEX]                    ;
                     00000000G    00           04  FB 0003F             CALLS   #4, LIB$ADDX                      ;
                                  24           50  E8 00046             BLBS    STATUS, 9$                        ;
                                                   04 00049             RET                                       ;
                                  50     08  BC42  D0 0004A 5$:         MOVL    @BUCKET[INDEX], R0                ; 1253
                          04      B3           50  D1 0004F             CMPL    R0, @4(R3)                        ;
```

```
                                                04  1E 00053          BGEQU   6$                                          ;
                                             50 04  B3 00055          MOVL    @4(R3), R0                                  ;
                                          08 BC42   50 00059 6$:      MOVL    R0, @BUCKET[INDEX]                          ;
                                                    0D 0005E          BRB     9$                                          ;
    04    A3        04    BC                10    00 00060 7$:      CMPZV   #0, #16, @BUFFER, 4(R3)                     ; 1259
                                                    04 00067          BNEQ    9$                                          ;
                                          08 BC42   D6 00069 8$:      INCL    @BUCKET[INDEX]                              ; 1260
                                             52 08  A3 0006D 9$:      ADDL2   8(R3), INDEX                                ; 1266
                                          9C    54    55 00071 10$:     AOBLSS  R5, I, 1$                               ; 1239
                                                50    01 00075          MOVL    #1, R0                                  ; 1269
                                                      04 00078          RET                                             ; 1270

; Routine Size:  121 bytes,    Routine Base:  CODE + 0132
```

```
;  441          1271  1 GLOBAL ROUTINE WRITE_SUMMARY (key, data) =
;  442          1272  1
;  443          1273  1 !----
;  444          1274  1 !
;  445          1275  1 ! Functional description
;  446          1276  1 !
;  447          1277  1 !       This routine is called to output summarization lines.
;  448          1278  1 !       These lines are accumulated and totals lines are output
;  449          1279  1 !       each time a key break occurs
;  450          1280  1 !
;  451          1281  1 ! Input parameters
;  452          1282  1 !
;  453          1283  1 !       KEY     = Address of descriptor of symbol .
;  454          1284  1 !       DATA    = Address of a summation bucket
;  455          1285  1 !
;  456          1286  1 !
;  457          1287  1 ! Output parameters
;  458          1288  1 !
;  459          1289  1 !       Any errors encountered are RETURNed immediately.
;  460          1290  1 !
;  461          1291  1 !----
;  462          1292  1
;  463          1293  2 BEGIN
;  464          1294  2
;  465          1295  2 EXTERNAL
;  466          1296  2         report_items,
;  467          1297  2         report_value: vector [],
;  468          1298  2         report_hdr1_fao,                ! Report header 1 FAO string
;  469          1299  2         report_hdr2_fao,                ! Report header 2 FAO string
;  470          1300  2         report_det_fao;                 ! Report detail FAO string
;  471          1301  2
;  472          1302  2 OWN
;  473          1303  2         title_desc:bblock [dsc$k_d_bln] ! Allocate dynamic descriptor
;  474          1304  2                 preset([dsc$b_class] = dsc$k_class_d),
;  475          1305  2         first: initial (true),          ! First time switch
;  476          1306  2         fao_list: VECTOR [MAX_REPORT+2];
;  477          1307  2
;  478          1308  2 LOCAL
;  479          1309  2         index_src,
;  480          1310  2         index_dst;
;  481          1311  2
;  482          1312  2 MAP
;  483          1313  2         report_hdr1_fao: bblock,        ! Upper report header
;  484          1314  2         report_hdr2_fao: bblock,        ! Lower report header
;  485          1315  2         key:  ref bblock [2],           ! Symbol descriptor
;  486          1316  2         data: ref vector [];            ! Summation record
;  487          1317  2
;  488          1318  2 If .first then
;  489          1319  3     BEGIN
;  490          1320  3     EXTERNAL first_date, last_date;
;  491          1321  3     LITERAL date1_size = 23, date2_size = 21;   ! Length of date strings
;  492          1322  3     LOCAL fill;
;  493          1323  3
;  494          1324  3     First = false;                          ! Clear switch
;  495          1325  3     SET_SCROLL (5, SCREEN (length));        ! Only effects Vt100's
;  496          1326  3     ERASE_PAGE ();                          ! Erase entire screen
;  497          1327  3     END;
```

```
;   498          1328  3      GET_VALUE ('TITLE', title_desc);                ! Get the title
;   499          1329  3      fill = SCREEN (width) - 2                       ! Compute available title area
;   500          1330  3          - date1_size - date2_size;
;   501          1331  3      if .title_desc [dsc$w_length] gtru .fill        ! If title is too long
;   502          1332  3          then BEGIN
;   503          1333  4              signal (acc$_titletrunc,1, .fill);      ! -- then warn user
;   504          1334  4              title_desc [dsc$w_length] = .fill;      ! -- truncate title
;   505          1335  4              END;
;   506          1336  3
;   507          1337  3      fill = .fill - .title_desc [dsc$w_length] + 2;
;   508          1338  3
;   509        P 1339  3      WRITE_LINE (XFAO (AD ('From: !17%D!#* !AS!#* To: !17%D'),
;   510        P 1340  3          first_date,                                 !  -"From" date
;   511        P 1341  3          .fill/2,                                    !  -fill to title
;   512        P 1342  3          title_desc,                                 !  -title
;   513        P 1343  3          .fill/2 + .fill mod 2,                      !  -fill after title
;   514          1344  3          last_date));                                !  -"To" Date
;   515          1345  3
;   516          1346  3      WRITE_LINE (XFAO (report_hdr1_fao));            ! Write report header 1
;   517          1347  3      WRITE_LINE (XFAO (report_hdr2_fao));            ! Write report header 2
;   518        P 1348  3      WRITE_LINE (XFAO (AD ('!#*-'),                  ! Underline them
;   519          1349  3          .report_hdr1_fao [dsc$w_length]));
;   520          1350  3      END;
;   521          1351  2
;   522          1352  2  fao_list [0] = .key;                               ! Store address of symbol desc
;   523          1353  2                                                     ! as first data item
;   524          1354  2  index_src = 1;
;   525          1355  2  index_dst = 1;
;   526          1356  2  Incr i to .report_items - 1 do
;   527          1357  2    BEGIN
;   528          1358  3    BIND rep_item = .report_value [.i]: vector [];
;   529          1359  3
;   530          1360  3  !
;   531          1361  3  ! Check the field length.  If 1 then copy the summation value. Else copy the
;   532          1362  3  ! summation value address.
;   533          1363  3  !
;   534          1364  3
;   535          1365  3    if .rep_item [sum_ent_bsize] eql 1
;   536          1366  3    then  fao_list [.index_dst] = .data [.index_src]
;   537          1367  4    else  BEGIN
;   538          1368  4          BUILTIN emul;
;   539          1369  4          lib$day (data [.index_src + 2], data [.index_src], data [.index_src]);
;   540          1370  4          emul (%ref(100000), data [.index_src], %ref(0), data [.index_src]);
;   541          1371  4          fao_list [.index_dst] = .data [.index_src + 2];
;   542          1372  4          index_dst = .index_dst + 1;
;   543          1373  4          fao_list [.index_dst] = data [.index_src];
;   544          1374  3          END;
;   545          1375  3    index_src = .index_src + .rep_item [sum_ent_bsize];
;   546          1376  3    index_dst = .index_dst + 1;
;   547          1377  2    END;
;   548          1378  2
;   549          1379  2  Write_line (XFAOL (report_det_fao, fao_list [0]));      ! Output a detail line
;   550          1380  2
;   551          1381  2  return true;
;   552          1382  1  END;
```

```
                                                                .PSECT  DATA,NOEXE,2

                                                        00# 00030 TITLE_DESC:
                                                                .BYTE   0[3]                                    ;
                                                        02  00033      .BYTE   2                                    ;
                                                            00034      .BLKB   4                                    ;
                                                 00000001   00038 FIRST:   .LONG   1                                    ;
                                                            0003C FAO_LIST:
                                                                .BLKB   128

                                                                .PSECT  CODE,NOWRT,2

                                                            001AB      .BLKB   1
                                00  00  00  45  4C  54  49  54  001AC P.AAB:   .ASCII  \TITLE\<0><0><0>                ;
                                                  00000005  001B4 P.AAA:   .LONG   5                                ;
                                                  00000000' 001B8          .ADDRESS P.AAB                          ;
20  2A  23  21  44  25  37  31  21  20  3A  6D  6F  72  46  001BC P.AAD:   .ASCII  \From: !17%D!#* !AS!#* To: !17%D\<0>  ;
25  37  31  21  20  3A  6F  54  20  2A  23  21  53  41  21  001CB                                                    ;
                                                        00  44  001DA                                               ;
                                                  0000001F  001DC P.AAC:   .LONG   31                               ;
                                                  00000000' 001E0          .ADDRESS P.AAD                          ;
                                            2D  2A  23  21  001E4 P.AAF:   .ASCII  \!#*-\                          ;
                                                  00000004  001E8 P.AAE:   .LONG   4                                ;
                                                  00000000' 001EC          .ADDRESS P.AAF                          ;

                                                                .EXTRN  REPORT_HDR1_FAO
                                                                .EXTRN  REPORT_HDR2_FAO
                                                                .EXTRN  REPORT_DET_FAO, FIRST_DATE
                                                                .EXTRN  LAST_DATE, SCR$SET_SCROLL
                                                                .EXTRN  SCREEN_CHAR, SCR$ERASE_PAGE
                                                                .EXTRN  CLI$GET_VALUE, SCR$PUT_LINE

                                                    0FFC 00000  .ENTRY  WRITE_SUMMARY, Save R2,R3,R4,R5,R6,R7,R8,-  ; 1271
                                                                R9,R10,R11                                          ;
                                    5B 00000000G    00   9E 00002      MOVAB   LIB$SYS_FAO, R11                    ;
                                    5A 00000000G    00   9E 00009      MOVAB   SCR$PUT_LINE, R10                   ;
                                    59 00000000'    EF   9E 00010      MOVAB   TITLE_DESC, R9                      ;
                                    5E      FDF8    CE   9E 00017      MOVAB   -520(SP), SP                        ;
                                    03        08    A9   E8 0001C      BLBS    FIRST, 1$                           ; 1318
                                                  014A  31 00020      BRW     7$                                  ;
                                                    08  A9   D4 00023 1$:    CLRL    FIRST                       ; 1324
                                    7E 00000000G    00   3C 00026      MOVZWL  SCREEN_CHAR+6, -(SP)                ; 1325
                                                        05   DD 0002D      PUSHL   #5                              ;
                00000000G           00               02   FB 0002F      CALLS   #2, SCR$SET_SCROLL                  ;
                                    52               50   D0 00036      MOVL    R0, STATUS                          ;
                                    0E               52   E9 00039      BLBC    STATUS, 2$                          ;
                                                     01   DD 0003C      PUSHL   #1                                  ; 1326
                                                     01   DD 0003E      PUSHL   #1                                  ;
                00000000G           00               02   FB 00040      CALLS   #2, SCR$ERASE_PAGE                  ;
                                    52               50   D0 00047      MOVL    R0, STATUS                          ;
                                    7C               52   E9 0004A 2$:    BLBC    STATUS, 4$                          ;
                                                     59   DD 0004D      PUSHL   R9                                  ; 1328
                                          FF71       CF   9F 0004F      PUSHAB  P.AAA                               ;
                00000000G           00               02   FB 00053      CALLS   #2, CLI$GET_VALUE                   ;
                                    52 00000000G    00   3C 0005A      MOVZWL  SCREEN_CHAR+4, FILL                 ; 1330
                                    52               2E   C2 00061      SUBL2   #46, FILL                           ;
```

```
                52          69          10              00 ED 00064          CMPZV   #0, #16, TITLE_DESC, FILL                       ; 1331
                                                        14 1B 00069          BLEQU   3$                                              ;
                                                        52 DD 0006B          PUSHL   FILL                                            ; 1333
                                                        01 DD 0006D          PUSHL   #1                                              ;
                                    00000000G           8F DD 0006F          PUSHL   #ACC$_TITLETRUNC                                ;
                        00000000G   00                  03 FB 00075          CALLS   #3, LIB$SIGNAL                                  ;
                                    69                  52 B0 0007C          MOVW    FILL, TITLE_DESC                                ; 1334
                                    50                  69 3C 0007F 3$:      MOVZWL  TITLE_DESC, R0                                  ; 1337
                50                  52                  50 C3 00082          SUBL3   R0, FILL, R0                                    ;
                                    52              02  A0 9E 00086          MOVAB   2(R0), FILL                                     ;
                                    7E              01  7D 0008A          MOVQ    #1, -(SP)                                       ; 1344
                            08  AE      0200       8F  3C 0008D          MOVZWL  #512, $$BUFFDESC                                ;
                            0C  AE        10       AE  9E 00093          MOVAB   $$BUFFER, $$BUFFDESC+4                          ;
                                    00000000G       00  9F 00098          PUSHAB  LAST_DATE                                       ;
                50                  52              02  C7 0009E          DIVL3   #2, FILL, R0                                    ;
                00                  52              01  7A 000A2          EMUL    #1, FILL, #0, -(SP)                             ;
                51  51              8E              02  7B 000A7          EDIV    #2, (SP)+, R1, R1                               ;
                                                6140    9F 000AC          PUSHAB  (R1)[R0]                                        ;
                                            0201        8F BB 000AF          PUSHR   #^M<R0,R9>                                      ;
                                    00000000G           00 9F 000B3          PUSHAB  FIRST_DATE                                      ;
                                              1C        AE 9F 000B9          PUSHAB  $$BUFFDESC                                      ;
                                              20        AE 9F 000BC          PUSHAB  $$BUFFDESC                                      ;
                                            FF29        CF 9F 000BF          PUSHAB  P.AAC                                           ;
                                    6B                  08 FB 000C3          CALLS   #8, LIB$SYS_FAO                                 ;
                                    52                  50 D0 000C6          MOVL    R0, STATUS                                      ;
                                    67                  52 E9 000C9 4$:      BLBC    STATUS, 5$                                      ;
                                              08        AE 9F 000CC          PUSHAB  $$BUFFDESC                                      ;
                                    6A                  03 FB 000CF          CALLS   #3, SCR$PUT_LINE                                ;
                                    52                  50 D0 000D2          MOVL    R0, STATUS                                      ;
                                    5B                  52 E9 000D5          BLBC    STATUS, 5$                                      ;
                                    7E                  01 7D 000D8          MOVQ    #1, -(SP)                                       ; 1346
                            08  AE      0200       8F  3C 000DB          MOVZWL  #512, $$BUFFDESC                                ;
                            0C  AE        10       AE  9E 000E1          MOVAB   $$BUFFER, $$BUFFDESC+4                          ;
                                              08        AE 9F 000E6          PUSHAB  $$BUFFDESC                                      ;
                                              0C        AE 9F 000E9          PUSHAB  $$BUFFDESC                                      ;
                                    00000000G           00 9F 000EC          PUSHAB  REPORT_HDR1_FAO                                 ;
                                    6B                  03 FB 000F2          CALLS   #3, LIB$SYS_FAO                                 ;
                                    52                  50 D0 000F5          MOVL    R0, STATUS                                      ;
                                    6C                  52 E9 000F8          BLBC    STATUS, 6$                                      ;
                                              08        AE 9F 000FB          PUSHAB  $$BUFFDESC                                      ;
                                    6A                  03 FB 000FE          CALLS   #3, SCR$PUT_LINE                                ;
                                    52                  50 D0 00101          MOVL    R0, STATUS                                      ;
                                    60                  52 E9 00104          BLBC    STATUS, 6$                                      ;
                                    7E                  01 7D 00107          MOVQ    #1, -(SP)                                       ; 1347
                            08  AE      0200       8F  3C 0010A          MOVZWL  #512, $$BUFFDESC                                ;
                            0C  AE        10       AE  9E 00110          MOVAB   $$BUFFER, $$BUFFDESC+4                          ;
                                              08        AE 9F 00115          PUSHAB  $$BUFFDESC                                      ;
                                              0C        AE 9F 00118          PUSHAB  $$BUFFDESC                                      ;
                                    00000000G           00 9F 0011B          PUSHAB  REPORT_HDR2_FAO                                 ;
                                    6B                  03 FB 00121          CALLS   #3, LIB$SYS_FAO                                 ;
                                    52                  50 D0 00124          MOVL    R0, STATUS                                      ;
                                    3D                  52 E9 00127          BLBC    STATUS, 6$                                      ;
                                              08        AE 9F 0012A          PUSHAB  $$BUFFDESC                                      ;
                                    6A                  03 FB 0012D          CALLS   #3, SCR$PUT_LINE                                ;
                                    52                  50 D0 00130          MOVL    R0, STATUS                                      ;
                                    31                  52 E9 00133 5$:      BLBC    STATUS, 6$                                      ;
                                    7E                  01 7D 00136          MOVQ    #1, -(SP)                                       ; 1349
```

```
            08    AE       0200   8F   3C 00139            MOVZWL  #512, $$BUFFDESC                        :
            0C    AE         10   AE   9E 0013F            MOVAB   $$BUFFER, $$BUFFDESC+4                  :
                  7E  00000000G   00   3C 00144            MOVZWL  REPORT_HDR1_FAO, -(SP)                  :
                             0C   AE   9F 0014B            PUSHAB  $$BUFFDESC                              :
                             10   AE   9F 0014E            PUSHAB  $$BUFFDESC                              :
                           FEA3   CF   9F 00151            PUSHAB  P.AAE                                   :
                  6B              04   FB 00155            CALLS   #4, LIB$SYS_FAO                         :
                  52              50   D0 00158            MOVL    R0, STATUS                              :
                  09              52   E9 0015B            BLBC    STATUS, 6$                              :
                             08   AE   9F 0015E            PUSHAB  $$BUFFDESC                              :
                  6A              03   FB 00161            CALLS   #3, SCR$PUT_LINE                        :
                  52              50   D0 00164            MOVL    R0, STATUS                              :
                  03              52   E8 00167  6$:       BLBS    STATUS, 7$                              :
                           009A   31      0016A            BRW     12$                                     :
            0C    A9         04   AC   D0 0016D  7$:       MOVL    KEY, FAO_LIST                           : 1352
                  54              01   D0 00172            MOVL    #1, INDEX_SRC                           : 1354
                  52              01   D0 00175            MOVL    #1, INDEX_DST                           : 1355
                  58  00000000G   00   D0 00178            MOVL    REPORT_ITEMS, R8                        : 1356
                  57              01   CE 0017F            MNEGL   #1, I                                   : 1366
                                  49   11 00182            BRB     11$                                     :
                  56  00000000G0047    D0 00184  8$:       MOVL    REPORT_VALUE[I], R6                     : 1358
                  55         08 BC44   DE 0018C            MOVAL   @DATA[INDEX_SRC], R5                    : 1366
                  01         08   A6   D1 00191            CMPL    8(R6), #1                               : 1365
                                  07   12 00195            BNEQ    9$                                      :
            0C A942               65   D0 00197            MOVL    (R5), FAO_LIST[INDEX_DST]               : 1366
                                  29   11 0019C            BRB     10$                                     :
                                  55   DD 0019E  9$:       PUSHL   R5                                      : 1369
                                  55   DD 001A0            PUSHL   R5                                      :
                  53         08 BC44   DE 001A2            MOVAL   @DATA[INDEX_SRC], R3                    :
                             08   A3   9F 001A7            PUSHAB  8(R3)                                   :
         00000000G  00            03   FB 001AA            CALLS   #3, LIB$DAY                             :
 65            00   65  000186A0   8F   7A 001B1            EMUL    #100000, (R5), #0, (R5)                 : 1370
            0C A942          08   A3   D0 001BA            MOVL    8(R3), FAO_LIST[INDEX_DST]              : 1371
                                  52   D6 001C0            INCL    INDEX_DST                               : 1372
            0C A942               55   D0 001C2            MOVL    R5, FAO_LIST[INDEX_DST]                 : 1373
                  54         08   A6   C0 001C7  10$:      ADDL2   8(R6), INDEX_SRC                        : 1375
                                  52   D6 001CB            INCL    INDEX_DST                               : 1376
       B3         57              58   F2 001CD  11$:      AOBLSS  R8, I, 8$                               : 1356
                  7E              01   7D 001D1            MOVQ    #1, -(SP)                               : 1379
            08    AE       0200   8F   3C 001D4            MOVZWL  #512, $$BUFFDESC                        :
            0C    AE         10   AE   9E 001DA            MOVAB   $$BUFFER, $$BUFFDESC+4                  :
                             0C   A9   9F 001DF            PUSHAB  FAO_LIST                                :
                             0C   AE   9F 001E2            PUSHAB  $$BUFFDESC                              :
                             10   AE   9F 001E5            PUSHAB  $$BUFFDESC                              :
                      00000000G   00   9F 001E8            PUSHAB  REPORT_DET_FAO                          :
         00000000G  00            04   FB 001EE            CALLS   #4, LIB$SYS_FAOL                        :
                  52              50   D0 001F5            MOVL    R0, STATUS                              :
                  0C              52   E9 001F8            BLBC    STATUS, 12$                             :
                             08   AE   9F 001FB            PUSHAB  $$BUFFDESC                              :
                  6A              03   FB 001FE            CALLS   #3, SCR$PUT_LINE                        :
                  52              50   D0 00201            MOVL    R0, STATUS                              :
                  0D              52   E8 00204            BLBS    STATUS, 13$                             :
                                  52   DD 00207  12$:      PUSHL   STATUS                                  :
         00000000G  00            01   FB 00209            CALLS   #1, LIB$SIGNAL                          :
                  50              52   D0 00210            MOVL    STATUS, R0                              :
                                       04 00213            RET                                             :
                  50              01   D0 00214  13$:      MOVL    #1, R0                                  : 1381
```

```
                                04 00217        RET                                     ; 1382

; Routine Size:  536 bytes,    Routine Base:  CODE + 01F0
```

```
;  554     1383  1 GLOBAL ROUTINE FIND_WATERMARK (BUFFER, BUCKET) =
;  555     1384  1
;  556     1385  1 !----
;  557     1386  1 !
;  558     1387  1 ! Functional description
;  559     1388  1 !
;  560     1389  1 !       This routine is called to scan the summarization buckets
;  561     1390  1 !       and determine the high watermark for each value.  A new
;  562     1391  1 !       summarization bucket is allocated and filled with the highest
;  563     1392  1 !       value encountered for each bucket item.
;  564     1393  1 !
;  565     1394  1 ! Input parameters
;  566     1395  1 !
;  567     1396  1 !       BUFFER  =       Address of a record to be summed
;  568     1397  1 !       BUCKET  =       Address of a summation bucket
;  569     1398  1 !
;  570     1399  1 ! Output parameters
;  571     1400  1 !
;  572     1401  1 !       Any errors encountered are RETURNed immediately.
;  573     1402  1 !
;  574     1403  1 !----
;  575     1404  1
;  576     1405  2 BEGIN
;  577     1406  2
;  578     1407  2 EXTERNAL
;  579     1408  2         bucket_size,                    ! Size of a summation bucket
;  580     1409  2         report_items,                   ! Number of report items
;  581     1410  2         report_value: vector [];        ! Table of pointers to offsets
;  582     1411  2
;  583     1412  2 MAP
;  584     1413  2         bucket: ref vector [],          ! Summary array
;  585     1414  2         buffer: ref bblock [];          ! Record buffer
;  586     1415  2
;  587     1416  2 LOCAL
;  588     1417  2         index;                          ! Index to bucket items
;  589     1418  2
;  590     1419  2 If .hwm_bucket eql 0                    ! If no bucket allocated yet
;  591     1420  3 then    BEGIN
;  592     1421  3         perform (allocate (.bucket_size, lwm_bucket));  ! Then get low bucket
;  593     1422  3         perform (allocate (.bucket_size, hwm_bucket));  ! and high bucket
;  594     1423  3         ch$fill (15, .bucket_size, .lwm_bucket);        ! Fill with high values
;  595     1424  3         ch$fill (0, .bucket_size, .hwm_bucket);         ! Fill with low values
;  596     1425  2         END;
;  597     1426  2
;  598     1427  2
;  599     1428  2 !
;  600     1429  2 ! For each report item, maximize the value stored in the watermark
;  601     1430  2 ! bucket with itself and the value stored in the current summation bucket.
;  602     1431  2 ! Minimize the value with the one stored in the lowwatermark bucket and
;  603     1432  2 ! store it.
;  604     1433  2 !
;  605     1434  2
;  606     1435  2 Index = 1;                              ! First bucket_item reserved
;  607     1436  2
;  608     1437  2 Incr i to .report_items - 1 do
;  609     1438  3   BEGIN
;  610     1439  3   BIND rep_item = .report_value [.i]: vector [];
```

```
;  611          1440  3
;  612          1441  !
;  613          1442  ! Check the field length.  If 1 then do a simple MAXU function to
;  614          1443  ! determine watermark.  Else assume length is 2 and do a COMPARE_QUAD.
;  615          1444  ! Do a MINU or COMPARE_QUAD for the minimum value
;  616          1445  !
;  617          1446
;  618          1447  3   if .rep_item [sum_ent_bsize] eql 1
;  619          1448  4   then  BEGIN
;  620          1449  4         hwm_bucket [.index] = maxu (.hwm_bucket [.index], .bucket [.index]);
;  621          1450  4         lwm_bucket [.index] = minu (.lwm_bucket [.index], .bucket [.index]);
;  622          1451  4         END
;  623          1452  4   else  BEGIN
;  624          1453  5         if COMPARE_QUAD (bucket [.index], GTRU, hwm_bucket [.index])
;  625          1454  4         then MOVE_QUAD (bucket [.index], hwm_bucket [.index]);
;  626          1455  4
;  627          1456  5         if COMPARE_QUAD (bucket [.index], LSSU, lwm_bucket [.index])
;  628          1457  4         then MOVE_QUAD (bucket [.index], lwm_bucket [.index]);
;  629          1458  3         END;
;  630          1459  3
;  631          1460  3   index = .index + .rep_item [sum_ent_bsize];
;  632          1461  2   END;
;  633          1462  2
;  634          1463  2 return true;
;  635          1464  1 END;


                                03FC 00000          .ENTRY  FIND_WATERMARK, Save R2,R3,R4,R5,R6,R7,R8,- ; 1383
                                                            R9
                            59 00000000G   00   9E 00002    MOVAB   ALLOCATE, R9
                            58 00000000G   00   9E 00009    MOVAB   BUCKET_SIZE, R8
                            57 00000000'   EF   9E 00010    MOVAB   HWM_BUCKET, R7
                                           67   D5 00017    TSTL    HWM_BUCKET                          1419
                                           27   12 00019    BNEQ    3$
                                     04    A7   9F 0001B    PUSHAB  LWM_BUCKET                          1421
                                           68   DD 0001E    PUSHL   BUCKET_SIZE
                            69             02   FB 00020    CALLS   #2, ALLOCATE
                            07             50   E9 00023    BLBC    STATUS, 1$
                                           57   DD 00026    PUSHL   R7                                  1422
                                           68   DD 00028    PUSHL   BUCKET_SIZE
                            69             02   FB 0002A    CALLS   #2, ALLOCATE
                            01             50   E8 0002D 1$: BLBS    STATUS, 2$
                                           04   00030       RET
                            56             68   D0 00031 2$: MOVL    BUCKET_SIZE, R6                    1423
                56          0F   6E        00   2C 00034    MOVC5   #0, (SP), #15, R6, @LWM_BUCKET
                                     04    B7   00039
                56          00   6E        00   2C 0003B    MOVC5   #0, (SP), #0, R6, @HWM_BUCKET      1424
                                     00    B7   00040
                            54             01   D0 00042 3$: MOVL    #1, INDEX                          1435
                            56             01   CE 00045    MNEGL   #1, I                              1450
                                           5D   11 00048    BRB     12$
                            55 00000000G0046   D0 0004A 4$: MOVL    REPORT_VALUE[I], R5                1439
                            51       00 B744   DE 00052    MOVAL   @HWM_BUCKET[INDEX], R1             1449
                            50       08 BC44   DE 00057    MOVAL   @BUCKET[INDEX], R0
```

```
                    53        04 B744   DE 0005C          MOVAL   @LWM_BUCKET[INDEX], R3                ; 1450
                    01        08   A5   D1 00061          CMPL    8(R5), #1                           ; 1447
                                 1E   12 00065          BNEQ    7$                                  ;
                    52             61   D0 00067          MOVL    (R1), R2                            ; 1449
                    60             52   D1 0006A          CMPL    R2, (R0)                            ;
                                   03   1E 0006D          BGEQU   5$                                  ;
                    52             60   D0 0006F          MOVL    (R0), R2                            ;
                    61             52   D0 00072 5$:      MOVL    R2, (R1)                            ;
                    52             63   D0 00075          MOVL    (R3), R2                            ; 1450
                    60             52   D1 00078          CMPL    R2, (R0)                            ;
                                   03   1B 0007B          BLEQU   6$                                  ;
                    52             60   D0 0007D          MOVL    (R0), R2                            ;
                    63             52   D0 00080 6$:      MOVL    R2, (R3)                            ;
                                   1E   11 00083          BRB     11$                                 ; 1447
                04  A1        04   A0   D1 00085 7$:      CMPL    4(R0), 4(R1)                        ; 1453
                                   03   12 0008A          BNEQ    8$                                  ;
                    61             60   D1 0008C          CMPL    (R0), (R1)                          ;
                                   03   1B 0008F 8$:      BLEQU   9$                                  ;
                    61             60   7D 00091          MOVQ    (R0), (R1)                          ; 1454
                04  A3        04   A0   D1 00094 9$:      CMPL    4(R0), 4(R3)                        ; 1456
                                   03   12 00099          BNEQ    10$                                 ;
                    63             60   D1 0009B          CMPL    (R0), (R3)                          ;
                                   03   1E 0009E 10$:      BGEQU   11$                                 ;
                    63             60   7D 000A0          MOVQ    (R0), (R3)                          ; 1457
                    54        08   A5   C0 000A3 11$:      ADDL2   8(R5), INDEX                        ; 1460
             9B         56 00000000G   00   F2 000A7 12$:      AOBLSS  REPORT_ITEMS, I, 4$                 ; 1437
                    50             01   D0 000AF          MOVL    #1, R0                              ; 1463
                                        04 000B2          RET                                         ; 1464

; Routine Size:  179 bytes,    Routine Base:  CODE + 0408
```

```
;  637          1465  1 GLOBAL ROUTINE WRITE_BAR_GRAPH (key, data) =
;  638          1466  1
;  639          1467  1 !----
;  640          1468  1 !
;  641          1469  1 ! Functional description
;  642          1470  1 !
;  643          1471  1 !       This routine is called to output summarization totals in
;  644          1472  1 !       the form of a bar graph.  For each invocation it outputs
;  645          1473  1 !       a summarization key and a horizontal bar.
;  646          1474  1 !
;  647          1475  1 !       The length of the bar is based on the number of available
;  648          1476  1 !       columns and the highest value found for the report item.
;  649          1477  1 !       The highest value has been previously obtained by FIND_WATERMARK
;  650          1478  1 !       and is available in a watermark_bucket built by that subroutine.
;  651          1479  1 !
;  652          1480  1 ! Input parameters
;  653          1481  1 !
;  654          1482  1 !       KEY     = Address of descriptor of symbol
;  655          1483  1 !       DATA    = Address of a summation bucket
;  656          1484  1 !
;  657          1485  1 ! Output parameters
;  658          1486  1 !
;  659          1487  1 !       Any errors encountered are RETURNed immediately.
;  660          1488  1 !
;  661          1489  1 !----
;  662          1490  1
;  663          1491  2 BEGIN
;  664          1492  2
;  665          1493  2 EXTERNAL
;  666          1494  2         report_value: vector [],
;  667          1495  2         report_hdr1_fao: bblock [],     ! First header for symbol column
;  668          1496  2         report_hdr2_fao: bblock [];     ! Second header for symbol column
;  669          1497  2
;  670          1498  2 GLOBAL
;  671          1499  2         reset_graph: initial (true);    ! Triggers graph initialization
;  672          1500  2
;  673          1501  2 OWN
;  674          1502  2         title_desc:bblock [dsc$k_d_bln] ! Allocate dynamic descriptor
;  675          1503  2                 preset([dsc$b_class] = dsc$k_class_d);
;  676          1504  2
;  677          1505  2 LOCAL
;  678          1506  2         worth;                          ! Worth of column
;  679          1507  2
;  680          1508  2 MAP
;  681          1509  2         key:  ref bblock [2],           ! Symbol descriptor
;  682          1510  2         data: ref vector [];            ! Summation record
;  683          1511  2
;  684          1512  2 If .reset_graph then
;  685          1513  3    BEGIN
;  686          1514  3    EXTERNAL first_date, last_date;
;  687          1515  3    LITERAL date1_size = 23, date2_size = 21;    ! Length of date strings
;  688          1516  3    LOCAL fill;
;  689          1517  3
;  690          1518  3    reset_graph = false;                         ! Clear switch
;  691          1519  3
;  692        P 1520  3    perform (setup_graph (
;  693        P 1521  3         .lwm_bucket [1],                        ! Minimum value
```

```
;   694          P 1522  3          .hwm_bucket [1],                        ! Maximum value
;   695          P 1523  3          .key [dsc$w_length]+columns_per_group,  ! Reserve left margin
;   696          P 1524  3          worth,                                  ! Receives computed worth
;   697            1525  3          desc));                                 ! Receives header string
;   698            1526  3
;   699            1527  3      GET_VALUE ('BAR_GRAPH', bar_char);           ! Get graphing character
;   700            1528  3
;   701            1529  3      perform (str$append (bar_fao, ad ('!132<!#*'')));! Load descriptor
;   702            1530  3      perform (str$append (bar_fao, bar_char));    ! Append to descriptor
;   703            1531  3      perform (str$append (bar_fao, ad ('!>'')));  ! Append to descriptor
;   704            1532  3      perform (str$append (report_hdr2_fao, desc));! Append graph header to symhdr
;   705            1533  3
;   706            1534  3      SET_SCROLL (5, SCREEN (length) - 1);         ! Only effects Vt100's
;   707            1535  3
;   708            1536  3      ERASE_PAGE ();                               ! Erase entire screen
;   709            1537  3
;   710            1538  3      GET_VALUE ('TITLE', title_desc);             ! Get the title
;   711            1539  3
;   712            1540  3      fill = SCREEN (width) - 2                    ! Compute available title area
;   713            1541  3           - date1_size - date2_size;
;   714            1542  3
;   715            1543  3      if .title_desc [dsc$w_length] gtru .fill     ! If title is too long
;   716            1544  4           then BEGIN
;   717            1545  4               signal (acc$_titletrunc,1, .fill);  ! -- then warn user
;   718            1546  4               title_desc [dsc$w_length] = .fill;  ! Truncate title if necessary
;   719            1547  4               END;
;   720            1548  3
;   721            1549  3      fill = .fill - .title_desc [dsc$w_length] + 2;
;   722            1550  3
;   723          P 1551  3      WRITE_LINE (XFAO (AD ('From: !17%D!#* !AS!#* To: !17%D'),
;   724          P 1552  3                  first_date,                      !  -"From" date
;   725          P 1553  3                  .fill72,                         !  -fill to title
;   726          P 1554  3                  title_desc,                      !  -title
;   727          P 1555  3                  .fill72 + .fill mod 2,           !  -fill after title
;   728            1556  3                  last_date));                     !  -"To" Date
;   729            1557  3
;   730            1558  3      WRITE_LINE (report_hdr1_fao);                ! Write report header 1
;   731            1559  3      WRITE_LINE (report_hdr2_fao);                ! Write report header 2
;   732          P 1560  3      WRITE_LINE (XFAO (AD ('!#*-'),               ! Underline it
;   733            1561  3           .report_hdr2_fao [dsc$w_length]));      !  -Header length
;   734            1562  3
;   735            1563  3      END;
;   736            1564  3
;   737          P 1565  2  Perform (lib$sys_fao (                           ! Build a bar
;   738          P 1566  2                  bar_fao,                         !  -FAO control string
;   739          P 1567  2                  0,                               !  -no output byte count
;   740          P 1568  2                  desc,                            !  -output buffer
;   741            1569  2                  .data [1] / .worth));            !  -length of bar
;   742            1570  2
;   743            1571  2  Incra ptr from .desc [dsc$a_pointer] to
;   744            1572  2          .desc [dsc$a_pointer] + .desc [dsc$w_length] -1 by columns_per_group
;   745            1573  2          do (.ptr)<0,8> = '!';
;   746            1574  2
;   747            1575  2  BEGIN
;   748            1576  2  BIND rep_item = .report_value [0]: vector [];
;   749            1577  3  if .rep_item [sum_ent_bsize] eql 1
;   750            1578  4  then    BEGIN
```

```
;       P    4              WRITE_LINE (XFAO (AD ('!#<!AS!9UL !AS!>'),      ! Write detail line
;       P    4                      .report_hdr2_fao [dsc$w_length],         !  -maximum line size
;       P    4                      .key,                                    !  -key descriptor
;       P    4                      .data [1],                               !  -data value
;            4                      desc));                                  !  -bar descriptor
;            4              END
;            4 else         BEGIN
;            4              OWN
;            4                      day,
;            4                      post_midnight: vector [2];
;            4              BUILTIN emul;
;            4              lib$day (day, data [1], post_midnight);
;            4              emul (%ref(100000), post_midnight, %ref(0), post_midnight);
;       P    4              WRITE_LINE (XFAO (AD ('!#<!AS!2UL !%T !AS!>'),  ! Write detail line
;       P    4                      .report_hdr2_fao [dsc$w_length],         !  -maximum line size
;       P    4                      .key,                                    !  -key descriptor
;       P    4                      .day,                                    !  -day value
;       P    4                      post_midnight,                           !  -time value
;            4                      desc));                                  !  -bar descriptor
;            3              END;
;            2 END;
;            2 return true;
;            1 END;



                                                                .PSECT  DATA,NOEXE,2

                                         00000001  000BC RESET_GRAPH::
                                                                .LONG   1                                  ;
                                              00#  000C0 TITLE_DESC:
                                                                .BYTE   0[3]                               ;
                                               02  000C3        .BYTE   2                                  ;
                                                   000C4        .BLKB   4
                                                   000C8 DAY:   .BLKB   4
                                                   000CC POST_MIDNIGHT:
                                                                .BLKB   8

                                                                .PSECT  CODE,NOWRT,2

                                                   004BB        .BLKB   1
         00  00  00  48  50  41  52  47  5F  52  41  42  004BC P.AAH:  .ASCII  \BAR_GRAPH\<0><0><0>         ;
                                         00000009  004C8 P.AAG:  .LONG   9                                  ;
                                         00000000' 004CC        .ADDRESS P.AAH                              ;
                         2A  23  21  3C  32  33  31  21  004D0 P.AAJ:  .ASCII  \!132<!#*\                  ;
                                         00000008  004D8 P.AAI:  .LONG   8                                  ;
                                         00000000' 004DC        .ADDRESS P.AAJ                              ;
                                     00  00  3E  21  004E0 P.AAL:  .ASCII  \!>\<0><0>                  ;
                                         00000002  004E4 P.AAK:  .LONG   2                                  ;
                                         00000000' 004E8        .ADDRESS P.AAL                              ;
                         00  00  00  45  4C  54  49  54  004EC P.AAN:  .ASCII  \TITLE\<0><0><0>             ;
                                         00000005  004F4 P.AAM:  .LONG   5                                  ;
                                         00000000' 004F8        .ADDRESS P.AAN                              ;
20  2A  23  21  44  25  37  31  21  20  3A  6D  6F  72  46  004FC P.AAP:  .ASCII  \From: !17%D!#* !AS!#* To: !17%D\<0>  ;
25  37  31  21  20  3A  6F  54  20  2A  23  21  53  41  21  0050B                                            ;
                                             00  44  0051A                                            ;
                                         0000001F  0051C P.AAO:  .LONG   31                                 ;
```

```
                                  00000000' 00520          .ADDRESS P.AAP                              ;
                         2D  2A  23  21  00524 P.AAR:   .ASCII  \!#*-\                                  ;
                                  00000004  00528 P.AAQ:   .LONG   4                                     ;
                                  00000000' 0052C          .ADDRESS P.AAR                              ;
21  53  41  21  20  4C  55  39  21  53  41  21  3C  23  21  00530 P.AAT:   .ASCII  \!#<!AS!9UL !AS!>\   ;
                                          3E  0053F                                                     ;
                                  00000010  00540 P.AAS:   .LONG   16                                    ;
                                  00000000' 00544          .ADDRESS P.AAT                              ;
20  54  25  21  20  4C  55  32  21  53  41  21  3C  23  21  00548 P.AAV:   .ASCII  \!#<!AS!2UL !%T !AS!>\ ;
                      3E  21  53  41  21  00557                                                         ;
                                  00000014  0055C P.AAU:   .LONG   20                                    ;
                                  00000000' 00560          .ADDRESS P.AAV                              ;


                                      07FC 00000          .ENTRY  WRITE_BAR_GRAPH, Save R2,R3,R4,R5,R6,R7,R8,-; 1465
                                                                  R9,R10                                 ;
             5A 00000000G  00  9E 00002          MOVAB   LIB$SIGNAL, R10                          ;
             59 00000000G  00  9E 00009          MOVAB   CLI$GET_VALUE, R9                        ;
             58 00000000G  00  9E 00010          MOVAB   STR$APPEND, R8                           ;
             57       FF49  CF  9E 00017          MOVAB   P.AAG, R7                                ;
             56 00000000G  00  9E 0001C          MOVAB   SCR$PUT_LINE, R6                         ;
             55 00000000G  00  9E 00023          MOVAB   LIB$SYS_FAO, R5                          ;
             54 00000000G  00  9E 0002A          MOVAB   REPORT_HDR2_FAO, R4                      ;
             53 00000000'  EF  9E 00031          MOVAB   DESC, R3                                 ;
             5E       FDF4  CE  9E 00038          MOVAB   -524(SP), SP                             ;
             03       009C  C3  E8 0003D          BLBS    RESET_GRAPH, 1$                          ; 1512
                      0163  31 00042          BRW     8$                                       ;
                      009C  C3  D4 00045 1$:      CLRL    RESET_GRAPH                              ; 1518
                             53  DD 00049          PUSHL   R3                                       ; 1525
                        04  AE  9F 0004B          PUSHAB  WORTH                                    ;
             7E         04  BC  3C 0004E          MOVZWL  @KEY, -(SP)                              ;
             6E             0F  C0 00052          ADDL2   #15, (SP)                                ;
             50         E0  A3  D0 00055          MOVL    HWM_BUCKET, R0                           ;
                        04  A0  DD 00059          PUSHL   4(R0)                                    ;
             50         E4  A3  D0 0005C          MOVL    LWM_BUCKET, R0                           ;
                        04  A0  DD 00060          PUSHL   4(R0)                                    ;
    0000V    CF             05  FB 00063          CALLS   #5, SETUP_GRAPH                          ;
             33             50  E9 00068          BLBC    STATUS, 2$                               ;
                        F8  A3  9F 0006B          PUSHAB  BAR_CHAR                                 ; 1527
                            57  DD 0006E          PUSHL   R7                                       ;
             69             02  FB 00070          CALLS   #2, CLI$GET_VALUE                        ;
                        10  A7  9F 00073          PUSHAB  P.AAI                                    ; 1529
                        F0  A3  9F 00076          PUSHAB  BAR_FAO                                  ;
             68             02  FB 00079          CALLS   #2, STR$APPEND                           ;
             1F             50  E9 0007C          BLBC    STATUS, 2$                               ;
                        F8  A3  9F 0007F          PUSHAB  BAR_CHAR                                 ; 1530
                        F0  A3  9F 00082          PUSHAB  BAR_FAO                                  ;
             68             02  FB 00085          CALLS   #2, STR$APPEND                           ;
             13             50  E9 00088          BLBC    STATUS, 2$                               ;
                        1C  A7  9F 0008B          PUSHAB  P.AAK                                    ; 1531
                        F0  A3  9F 0008E          PUSHAB  BAR_FAO                                  ;
             68             02  FB 00091          CALLS   #2, STR$APPEND                           ;
             07             50  E9 00094          BLBC    STATUS, 2$                               ;
                            53  DD 00097          PUSHL   R3                                       ; 1532
                            54  DD 00099          PUSHL   R4                                       ;
             68             02  FB 0009B          CALLS   #2, STR$APPEND                           ;
```

```
                                01              50  E8 0009E 2$:    BLBS    STATUS, 3$                                      ;
                                                    04 000A1        RET                                                     ;
                                7E 00000000G    00  3C 000A2 3$:    MOVZWL  SCREEN_CHAR+6, -(SP)                            ; 1534
                                                6E  D7 000A9        DECL    (SP)                                            ;
                                                05  DD 000AB        PUSHL   #5                                              ;
                        00000000G 00            02  FB 000AD        CALLS   #2, SCR$SET_SCROLL                              ;
                                  52            50  D0 000B4        MOVL    R0, STATUS                                      ;
                                  0E            52  E9 000B7        BLBC    STATUS, 4$                                      ;
                                                01  DD 000BA        PUSHL   #1                                              ; 1536
                                                01  DD 000BC        PUSHL   #1                                              ;
                        00000000G 00            02  FB 000BE        CALLS   #2, SCR$ERASE_PAGE                              ;
                                  52            50  D0 000C5        MOVL    R0, STATUS                                      ;
                                  7C            52  E9 000C8 4$:    BLBC    STATUS, 6$                                      ;
                                        00A0    C3  9F 000CB        PUSHAB  TITLE_DESC                                      ; 1538
                                          2C    A7  9F 000CF        PUSHAB  P.AAM                                           ;
                                  69            02  FB 000D2        CALLS   #2, CLI$GET_VALUE                               ;
                                  52 00000000G  00  3C 000D5        MOVZWL  SCREEN_CHAR+4, FILL                             ; 1541
                                  52            2E  C2 000DC        SUBL2   #46, FILL                                       ;
      52        00A0    C3      10            00  ED 000DF        CMPZV   #0, #16, TITLE_DESC, FILL                       ; 1543
                                                12  1B 000E6        BLEQU   5$                                              ;
                                                52  DD 000E8        PUSHL   FILL                                            ; 1545
                                                01  DD 000EA        PUSHL   #1                                              ;
                                00000000G       8F  DD 000EC        PUSHL   #ACC$_TITLETRUNC                                ;
                                  6A            03  FB 000F2        CALLS   #3, LIB$SIGNAL                                  ;
                          00A0    C3            52  B0 000F5        MOVW    FILL, TITLE_DESC                                ; 1546
                                  50      00A0  C3  3C 000FA 5$:    MOVZWL  TITLE_DESC, R0                                  ; 1549
                    50            52            50  C3 000FF        SUBL3   R0, FILL, R0                                    ;
                                  52          02  A0 9E 00103      MOVAB   2(R0), FILL                                     ;
                                  7E          01  7D 00107        MOVQ    #1, -(SP)                                       ; 1556
                              0C  AE        0200  8F 3C 0010A      MOVZWL  #512, $$BUFFDESC                                ;
                              10  AE          14  AE 9E 00110      MOVAB   $$BUFFER, $$BUFFDESC+4                          ;
                                00000000G     00  9F 00115        PUSHAB  LAST_DATE                                       ;
                    50            52            02  C7 0011B        DIVL3   #2, FILL, R0                                    ;
      7E            00            52            01  7A 0011F        EMUL    #1, FILL, #0, -(SP)                             ;
      51            51            8E            02  7B 00124        EDIV    #2, (SP)+, R1, R1                               ;
                                              6140  9F 00129        PUSHAB  (R1)[R0]                                        ;
                                        00A0    C3  9F 0012C        PUSHAB  TITLE_DESC                                      ;
                                                50  DD 00130        PUSHL   R0                                              ;
                                00000000G       00  9F 00132        PUSHAB  FIRST_DATE                                      ;
                                          20    AE  9F 00138        PUSHAB  $$BUFFDESC                                      ;
                                          24    AE  9F 0013B        PUSHAB  $$BUFFDESC                                      ;
                                          54    A7  9F 0013E        PUSHAB  P.AAO                                           ;
                                  65            08  FB 00141        CALLS   #8, LIB$SYS_FAO                                 ;
                                  52            50  D0 00144        MOVL    R0, STATUS                                      ;
                                  58            52  E9 00147 6$:    BLBC    STATUS, 7$                                      ;
                                          0C    AE  9F 0014A        PUSHAB  $$BUFFDESC                                      ;
                                  66            03  FB 0014D        CALLS   #3, SCR$PUT_LINE                                ;
                                  52            50  D0 00150        MOVL    R0, STATUS                                      ;
                                  4C            52  E9 00153        BLBC    STATUS, 7$                                      ;
                                  7E            01  7D 00156        MOVQ    #1, -(SP)                                       ; 1558
                                00000000G       00  9F 00159        PUSHAB  REPORT_HDR1_FAO                                 ;
                                  66            03  FB 0015F        CALLS   #3, SCR$PUT_LINE                                ;
                                  52            50  D0 00162        MOVL    R0, STATUS                                      ;
                                  3A            52  E9 00165        BLBC    STATUS, 7$                                      ;
                                  7E            01  7D 00168        MOVQ    #1, -(SP)                                       ; 1559
                                                54  DD 0016B        PUSHL   R4                                              ;
                                  66            03  FB 0016D        CALLS   #3, SCR$PUT_LINE                                ;
```

```
                        52              50  D0 00170          MOVL    R0, STATUS                                      ;
                        2C              52  E9 00173          BLBC    STATUS, 7$                                      ;
                        7E              01  7D 00176          MOVQ    #1, -(SP)                                       ; 1561
                    0C  AE        0200  8F  3C 00179          MOVZWL  #512, $$BUFFDESC                                ;
                    10  AE          14  AE  9E 0017F          MOVAB   $$BUFFER, $$BUFFDESC+4                          ;
                        7E              64  3C 00184          MOVZWL  REPORT_HDR2_FAO, -(SP)                          ;
                                    10  AE  9F 00187          PUSHAB  $$BUFFDESC                                      ;
                                    14  AE  9F 0018A          PUSHAB  $$BUFFDESC                                      ;
                                    60  A7  9F 0018D          PUSHAB  P.AAQ                                           ;
                                        65  04 FB 00190          CALLS   #4, LIB$SYS_FAO                             ;
                                        52  50 D0 00193          MOVL    R0, STATUS                                  ;
                                        09  52 E9 00196          BLBC    STATUS, 7$                                  ;
                                            0C  AE  9F 00199          PUSHAB  $$BUFFDESC                              ;
                                        66  03 FB 0019C          CALLS   #3, SCR$PUT_LINE                            ;
                                        52  50 D0 0019F          MOVL    R0, STATUS                                  ;
                                        03  52 E8 001A2  7$:      BLBS    STATUS, 8$                                  ;
                                          00BC  31 001A5          BRW     14$                                         ;
                                        52  08  AC D0 001A8  8$:      MOVL    DATA, R2                                ; 1569
                        7E          04  A2  6E C7 001AC          DIVL3   WORTH, 4(R2), -(SP)                         ;
                                            53  DD 001B1          PUSHL   R3                                          ;
                                            7E  D4 001B3          CLRL    -(SP)                                       ;
                                            F0  A3  9F 001B5          PUSHAB  BAR_FAO                                 ;
                                        65  04 FB 001B8          CALLS   #4, LIB$SYS_FAO                             ;
                                        01  50 E8 001BB          BLBS    STATUS, 9$                                  ;
                                            04 001BE          RET                                                     ;
                                        50  63 3C 001BF  9$:      MOVZWL  DESC, R0                                    ; 1572
                                        50  04  A3 C0 001C2          ADDL2   DESC+4, R0                              ;
                                        50  D7 001C6          DECL    R0                                          ;
                                        51  04  A3 D0 001C8          MOVL    DESC+4, PTR                             ; 1571
                                        07  11 001CC          BRB     11$                                         ;
                                    81  7C  8F 90 001CE  10$:      MOVB    #124, (PTR)+                            ; 1573
                                    51      0E C0 001D2          ADDL2   #14, PTR                                    ;
                                    50      51 D1 001D5  11$:      CMPL    PTR, R0                                 ;
                                            F4 1B 001D8          BLEQU   10$                                         ;
                                        50 00000000G  00 D0 001DA          MOVL    REPORT_VALUE, R0                    ; 1576
                                04  AE      0200  8F 3C 001E1          MOVZWL  #512, $$BUFFDESC                        ; 1583
                                01      08  A0 D1 001E7          CMPL    8(R0), #1                                   ; 1577
                                            21 12 001EB          BNEQ    12$                                         ;
                                        7E      01 7D 001ED          MOVQ    #1, -(SP)                               ; 1583
                                10  AE      14  AE 9E 001F0          MOVAB   $$BUFFER, $$BUFFDESC+4                  ;
                                            53  DD 001F5          PUSHL   R3                                          ;
                                            04  A2 DD 001F7          PUSHL   4(R2)                                   ;
                                            04  AC DD 001FA          PUSHL   KEY                                     ;
                                        7E      64 3C 001FD          MOVZWL  REPORT_HDR2_FAO, -(SP)                  ;
                                            1C  AE 9F 00200          PUSHAB  $$BUFFDESC                              ;
                                            20  AE 9F 00203          PUSHAB  $$BUFFDESC                              ;
                                            78  A7 9F 00206          PUSHAB  P.AAS                                   ;
                                        65      07 FB 00209          CALLS   #7, LIB$SYS_FAO                         ;
                                            44 11 0020C          BRB     13$                                         ;
                                        00AC  C3 9F 0020E  12$:      PUSHAB  POST_MIDNIGHT                           ; 1590
                                            04  A2 9F 00212          PUSHAB  4(R2)                                   ;
                                        00A8  C3 9F 00215          PUSHAB  DAY                                     ;
                            00000000G   00      03 FB 00219          CALLS   #3, LIB$DAY                             ;
00AC    C3              00      00AC  C3 000186A0  8F 7A 00220          EMUL    #100000, POST_MIDNIGHT, #0, POST_MIDNIGHT ; 1591
                                        7E      01 7D 0022D          MOVQ    #1, -(SP)                               ; 1597
                                10  AE      14  AE 9E 00230          MOVAB   $$BUFFER, $$BUFFDESC+4                  ;
                                            53  DD 00235          PUSHL   R3                                          ;
```

```
                00AC   C3  9F 00237          PUSHAB  POST_MIDNIGHT                                   ;
                00A8   C3  DD 0023B          PUSHL   DAY                                             ;
                   04  AC  DD 0023F          PUSHL   KEY                                             ;
             7E        64  3C 00242          MOVZWL  REPORT_HDR2_FAO, -(SP)                          ;
                   20  AE  9F 00245          PUSHAB  $$BUFFDESC                                      ;
                   24  AE  9F 00248          PUSHAB  $$BUFFDESC                                      ;
                0094   C7  9F 0024B          PUSHAB  P.AAU                                           ;
             65        08  FB 0024F          CALLS   #8, LIB$SYS_FAO                                 ;
             52        50  D0 00252 13$:      MOVL    R0, STATUS                                      ;
             0C        52  E9 00255          BLBC    STATUS, 14$                                     ;
                   0C  AE  9F 00258          PUSHAB  $$BUFFDESC                                      ;
             66        03  FB 0025B          CALLS   #3, SCR$PUT_LINE                                ;
             52        50  D0 0025E          MOVL    R0, STATUS                                      ;
             09        52  E8 00261          BLBS    STATUS, 15$                                     ;
                       52  DD 00264 14$:      PUSHL   STATUS                                          ;
             6A        01  FB 00266          CALLS   #1, LIB$SIGNAL                                  ;
             50        52  D0 00269          MOVL    STATUS, R0                                      ;
                           04 0026C          RET                                                     ;
             50        01  D0 0026D 15$:      MOVL    #1, R0                                          ; 1600
                           04 00270          RET                                                     ; 1601

; Routine Size:  625 bytes,    Routine Base:  CODE + 0564
```

```
;  775     1602  1 ROUTINE SETUP_GRAPH (MINVAL, MAXVAL, RESERVE, WORTH, DESC) =
;  776     1603  1
;  777     1604  1 !----
;  778     1605  1 !
;  779     1606  1 ! Functional description
;  780     1607  1 !
;  781     1608  1 !       This routine calculates the worth of a column for a bar
;  782     1609  1 !       graph and prepares a header line to go over the graph.
;  783     1610  1 !
;  784     1611  1 ! Input parameters
;  785     1612  1 !
;  786     1613  1 !       MINVAL  = Minimum value to be graphed
;  787     1614  1 !       MAXVAL  = Maximum value to be graphed
;  788     1615  1 !       RESERVE = Number of columns to reserve for symbols
;  789     1616  1 !
;  790     1617  1 ! Output parameters
;  791     1618  1 !
;  792     1619  1 !       WORTH   = Address of a longword to receive worth of a column
;  793     1620  1 !       DESC    = Address of a dynamic descriptor to receive header
;  794     1621  1 !
;  795     1622  1 !       Any errors encountered are RETURNed immediately.
;  796     1623  1 !
;  797     1624  1 !----
;  798     1625  1
;  799     1626  2 BEGIN
;  800     1627  2
;  801     1628  2 LOCAL
;  802     1629  2         min,                            ! Holds local minimum value
;  803     1630  2         width,                          ! Width of output device
;  804     1631  2         range,                          ! maxval - minval
;  805     1632  2         numgrp;                         ! Number of groups
;  806     1633  2
;  807     1634  2
;  808     1635  2 Width = SCREEN (width) - .reserve;      ! Calc useable space
;  809     1636  2
;  810     1637  2 Width = (.width/columns_per_group) *     ! Use integral number of groups
;  811     1638  2         columns_per_group;
;  812     1639  2
;  813     1640  2 If .minval lssu (.maxval/4) then min = 0
;  814     1641  2    else min = .minval;                  ! Only offset if significant
;  815     1642  2
;  816     1643  2 Range = maxu (.maxval - .min, 1);       ! Compute range of graph
;  817     1644  2
;  818     1645  2 .Worth = (.range+.width)/.width;        ! Calc worth of column
;  819     1646  2
;  820     1647  2 Numgrp = .width / columns_per_group;    ! Calc number of groups
;  821     1648  2
;  822     1649  2 Perform (str$dupl_char (.desc, 0));     ! Zero result descriptor
;  823     1650  2
;  824     1651  2 Incru i to .width by columns_per_group do   ! For each group
;  825     1652  3    BEGIN
;  826   P 1653  3    perform (lib$sys_fao (                  ! Create a detail header
;  827   P 1654  3          ad ('!#<!UL!>'),                  !  -control string
;  828   P 1655  3          0,                                !  -no result length needed
;  829   P 1656  3          wdesc,                            !  -output buffer
;  830   P 1657  3          columns_per_group,                !  -width of a column group
;  831     1658  3          .min + (.i * ..worth)));          !  -group value
```

```
;  832        1659  3
;  833      P 1660  3       perform (str$append (                         ! Concatenate with header
;  834      P 1661  3           .desc,                                    !  -destination
;  835        1662  3           wdesc));                                  !  -source
;  836        1663  2       END;
;  837        1664  2
;  838        1665  2   Perform (str$left (.desc, .desc, width));         ! Limit max string size
;  839        1666  2
;  840        1667  2   Return true;
;  841        1668  1   END;


                                             007D5          .BLKB    3
           3E  21  4C  55  21  3C  23  21   007D8 P.AAX:   .ASCII   \!#<!UL!>\                          ;
                                 00000008   007E0 P.AAW:   .LONG    8                                   ;
                                 00000000'  007E4          .ADDRESS P.AAX                               ;


                                     001C   00000 SETUP_GRAPH:
                                                           .WORD    Save R2,R3,R4                       ; 1602
                       54 00000000'    EF   9E 00002       MOVAB    WDESC, R4                           ;
                       50 00000000G    00   3C 00009       MOVZWL   SCREEN_CHAR+4, R0                   ; 1635
           7E          50          0C  AC   C3 00010       SUBL3    RESERVE, R0, WIDTH                  ;
           50          6E              0F   C7 00015       DIVL3    #15, WIDTH, R0                      ; 1637
           6E          50              0F   C5 00019       MULL3    #15, R0, WIDTH                      ;
           50      08  AC              04   C7 0001D       DIVL3    #4, MAXVAL, R0                      ; 1640
                       50          04  AC   D1 00022       CMPL     MINVAL, R0                          ;
                                       04   1E 00026       BGEQU    1$                                  ;
                                       53   D4 00028       CLRL     MIN                                 ;
                                       04   11 0002A       BRB      2$                                  ;
                       53          04  AC   D0 0002C 1$:   MOVL     MINVAL, MIN                         ; 1641
           50      08  AC              53   C3 00030 2$:   SUBL3    MIN, MAXVAL, R0                     ; 1643
                                       03   12 00035       BNEQ     3$                                  ;
                       50              01   D0 00037       MOVL     #1, R0                              ;
                       50              6E   C0 0003A 3$:   ADDL2    WIDTH, R0                           ; 1645
      10   BC          50              6E   C7 0003D       DIVL3    WIDTH, R0, @WORTH                   ;
           50          6E              0F   C7 00042       DIVL3    #15, WIDTH, NUMGRP                  ; 1647
                                       7E   D4 00046       CLRL     -(SP)                               ; 1649
                                   14  AC   DD 00048       PUSHL    DESC                                ;
              00000000G  00            02   FB 0004B       CALLS    #2, STR$DUPL_CHAR                   ;
                         4B            50   E9 00052       BLBC     STATUS, 6$                          ;
                                       52   D4 00055       CLRL     I                                   ; 1651
                                       2D   11 00057       BRB      5$                                  ;
           50          52          10  BC   C5 00059 4$:   MULL3    @WORTH, I, R0                       ; 1658
                                     6043   9F 0005E       PUSHAB   (R0)[MIN]                           ;
                                       0F   DD 00061       PUSHL    #15                                 ;
                                       54   DD 00063       PUSHL    R4                                  ;
                                       7E   D4 00065       CLRL     -(SP)                               ;
                                   8E  AF   9F 00067       PUSHAB   P.AAW                               ;
              00000000G  00            05   FB 0006A       CALLS    #5, LIB$SYS_FAO                     ;
                         2C            50   E9 00071       BLBC     STATUS, 6$                          ;
                                       54   DD 00074       PUSHL    R4                                  ; 1662
                                   14  AC   DD 00076       PUSHL    DESC                                ;
              00000000G  00            02   FB 00079       CALLS    #2, STR$APPEND                      ;
                         1D            50   E9 00080       BLBC     STATUS, 6$                          ;
```

```
                          52              0F  C0 00083          ADDL2   #15, I                                  ; 1651
                          6E              52  D1 00086 5$:     CMPL    I, WIDTH                                ;
                                          CE  1B 00089         BLEQU   4$                                      ;
                                          5E  DD 0008B         PUSHL   SP                                      ; 1665
                                  14      AC  DD 0008D         PUSHL   DESC                                    ;
                                  14      AC  DD 00090         PUSHL   DESC                                    ;
                  00000000G       00          03  FB 00093     CALLS   #3, STR$LEFT                            ;
                                  03          50  E9 0009A     BLBC    STATUS, 6$                              ;
                                  50          01  D0 0009D     MOVL    #1, R0                                  ; 1667
                                              04  00A0 6$:     RET                                             ; 1668

; Routine Size:  161 bytes,    Routine Base:  CODE + 07E8
```

```
;  843          1669  1 GLOBAL ROUTINE WRITE_TOTALS  =
;  844          1670  1
;  845          1671  1 !----
;  846          1672  1 !
;  847          1673  1 ! Functional description
;  848          1674  1 !
;  849          1675  1 !       This routine is called when a totals page is required.  A
;  850          1676  1 !       totals page contains statistics accumlated during the running
;  851          1677  1 !       of the program, such as number of records read.
;  852          1678  1 !
;  853          1679  1 ! Input parameters
;  854          1680  1 !
;  855          1681  1 !
;  856          1682  1 ! Output parameters
;  857          1683  1 !
;  858          1684  1 !       Any errors encountered are RETURNed immediately.
;  859          1685  1 !
;  860          1686  1 !----
;  861          1687  1
;  862          1688  2 BEGIN
;  863          1689  2 return true;
;  864          1690  1 END;


                                0000 00000          .ENTRY  WRITE_TOTALS, Save nothing                      ; 1669
                          50    01   D0 00002          MOVL    #1, R0                                          ; 1689
                                     04 00005          RET                                                     ; 1690

; Routine Size:  6 bytes,    Routine Base:  CODE + 0889
```

```
;  866          1691  1 END
;  867          1692  0 ELUDOM



                                                    .EXTRN  LIB$SIGNAL

;                               PSECT SUMMARY

;       Name                    Bytes                   Attributes

;  DATA                          212  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  CODE                         2191  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         36           0        581        00:01.1
;  _$255$DUA28:[SYSLIB]TPAMAC.L32;1     42          0           0         14        00:00.1



;                                   COMMAND QUALIFIERS

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

; Size:          1934 code + 469 data bytes
; Run Time:        00:41.9
; Elapsed Time:    01:21.8
; Lines/CPU Min:    2420
; Lexemes/CPU-Min: 26755
; Memory Used:  273 pages
; Compilation Complete
```

AEDCLENUP
LIS

SYMBOLS
LIS

UTILITY
LIS

ACLEDTDEF
REQ

ACLEDT

SETSHOACL
MAP

ACLEDT
MAP

AEDLCLDEE
SDL

SUMMARY
LIS

SYMBOLMSG
LIS

AEDDECODE
LIS