```
   SSSSSSSSSSS  UUU          UUU  MMM          MMM
   SSSSSSSSSSS  UUU          UUU  MMM          MMM
   SSSSSSSSSSS  UUU          UUU  MMM          MMM
SSS             UUU          UUU  MMMMMMM    MMMMMM
SSS             UUU          UUU  MMMMMMM    MMMMMM
SSS             UUU          UUU  MMMMMMM    MMMMMM
SSS             UUU          UUU  MMM MMM MMM MMM
SSS             UUU          UUU  MMM  MMM  MMM MMM
SSS             UUU          UUU  MMM   M   MMM MMM
   SSSSSSSSS    UUU          UUU  MMM       MMM MMM
   SSSSSSSSS    UUU          UUU  MMM       MMM MMM
   SSSSSSSSS    UUU          UUU  MMM       MMM MMM
            SSS UUU          UUU  MMM       MMM MMM
            SSS UUU          UUU  MMM       MMM MMM
            SSS UUU          UUU  MMM       MMM MMM
            SSS UUU          UUU  MMM       MMM MMM
            SSS UUU          UUU  MMM       MMM MMM
            SSS UUU          UUU  MMM       MMM MMM
SSSSSSSSSSS     UUUUUUUUUUUUUUUU  MMM       MMM MMM
SSSSSSSSSSS     UUUUUUUUUUUUUUUU  MMM       MMM MMM
SSSSSSSSSSS     UUUUUUUUUUUUUUUU  MMM       MMM MMM
SSSSSSSSSSS     UUUUUUUUUUUUUUUU  MMM       MMM MMM
```

```
UU      UU  PPPPPPPP    DDDDDDDD      AAAAAA    TTTTTTTTTT  EEEEEEEEEE
UU      UU  PPPPPPPPP   DDDDDDDDD     AAAAAA    TTTTTTTTTT  EEEEEEEEEE
UU      UU  PP      PP  DD      DD   AA    AA      TT      EE
UU      UU  PP      PP  DD      DD   AA    AA      TT      EE
UU      UU  PP      PP  DD      DD   AA    AA      TT      EE
UU      UU  PP      PP  DD      DD   AA    AA      TT      EE
UU      UU  PPPPPPPPP   DD      DD   AA    AA      TT      EEEEEEEE
UU      UU  PPPPPPPP    DD      DD   AA    AA      TT      EEEEEEEE
UU      UU  PP          DD      DD   AAAAAAAAAA    TT      EE
UU      UU  PP          DD      DD   AAAAAAAAAA    TT      EE
UU      UU  PP          DD      DD   AA    AA      TT      EE                  ....
UU      UU  PP          DD      DD   AA    AA      TT      EE                  ....
UUUUUUUUUU  PP          DDDDDDDDD    AA    AA      TT      EEEEEEEEEE          ....
UUUUUUUUUU  PP          DDDDDDDD     AA    AA      TT      EEEEEEEEEE          ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

Table of contents

```
0000     1          .TITLE  UPDATE
0000     2          .IDENT  'V04-000'
0000     3  ;
0000     4  ;*****************************************************************************
0000     5  ;*                                                                           *
0000     6  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     7  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     8  ;*  ALL RIGHTS RESERVED.                                                     *
0000     9  ;*                                                                           *
0000    10  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
0000    11  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
0000    12  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
0000    13  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
0000    14  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
0000    15  ;*  TRANSFERRED.                                                             *
0000    16  ;*                                                                           *
0000    17  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
0000    18  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0000    19  ;*  CORPORATION.                                                             *
0000    20  ;*                                                                           *
0000    21  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
0000    22  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPFLIED BY DIGITAL.                  *
0000    23  ;*                                                                           *
0000    24  ;*                                                                           *
0000    25  ;*****************************************************************************
0000    26  ;
0000    27  ;
0000    28  ;++
0000    29  ;
0000    30  ; Facility:
0000    31  ;
0000    32  ;       SUMSLP utility
0000    33  ;
0000    34  ; Environment:
0000    35  ;
0000    36  ;       User mode
0000    37  ;
0000    38  ; Author:
0000    39  ;
0000    40  ;       R. Newland      18-Apr-1979
0000    41  ;
0000    42  ; Modified by:
0000    43  ;
0000    44  ;       V03-001 BLS0173         Benn Schreiber          26-MAY-1982
0000    45  ;               Use general addressing mode for SUM$XXX_ERR calls.
0000    46  ;--
```

DECLARATIONS

```
                     0000     48          .SBTTL  DECLARATIONS
                     0000     49 ;
                     0000     50 ;
                     0000     51          $FABDEF                         ; RMS FAB definitions
                     0000     52          $RABDEF                         ; RMS RAB definitions
                     0000     53          $NAMDEF                         ; RMS NAM definitions
                     0000     54          DEFSSLGEN                       ; Define SUMSLP general values
                     0000     55          DEFSSLFLG                       ; Define SUMSLP flags
                     0000     56          DEFSUMFLG                       ; Define SUM flags
                     0000     57          DEFSUMCBL                       ; Define SUM control block offsets
                     0000     58          DEFSSLRHB                       ; Define SUM record header buffer
                     0000     59          $STSDEF                         ; Define status fields
                     0000     60          $SSDEF                          ; Define system status codes
                     0000     61 ;
                     0000     62 ;
0000000C             0000     63          FF = ^XC                        ; Form-feed character
                     0000     64
                00000000     65          .PSECT  SUM$CODE,EXE,NOWRT
```

```
                        67          .SBTTL  UPDATE
                        68 ;
                        69 ;++
                        70 ;
                        71 ; Functional description:
                        72 ;
                        73 ; This routine is called to perform the update operation and acts
                        74 ; as the main line code when applying update packets to an input file.
                        75 ;
                        76 ;
                        77 ; Calling sequence:
                        78 ;
                        79 ;       BSB     SUM$UPDATE
                        80 ;
                        81 ;
                        82 ; Input paramters:
                        83 ;
                        84 ;       None
                        85 ;
                        86 ;
                        87 ; Implicit inputs:
                        88 ;
                        89 ;       SUM$AX_INPUTRAB = Input file RAB
                        90 ;       SUM$GL_UPDATES  = Address of update files list
                        91 ;       SUM$AX_CBL      = Address of SUM control block
                        92 ;
                        93 ;
                        94 ; Output parameters:
                        95 ;
                        96 ;       None
                        97 ;
                        98 ;
                        99 ; Implicit outputs:
                       100 ;
                       101 ;       None
                       102 ;
                       103 ;
                       104 ; Side effects:
                       105 ;
                       106 ;       None
                       107 ;
                       108 ;--
                       109 ;
                       110 SUM$UPDATE::
                       111 ;
5B   0000'CF   DE      112         MOVAL   W^SUM$GL_FLAGS,R11      ; Set flags longword address
                       113 ;
                       114 ; Open all input files (including any update files) all required
                       115 ; output files.
                       116 ;
     0065      30      117         BSBW    OPEN_FILES              ; Open and create all files
    61 50      E9      118         BLBC    R0,40$                  ; Error if LBC
                       119 ;
                       120 ;
                       121 ; Perform initial processing of update files
                       122 ;
0000'CF    9F          123         PUSHAB  W^SUM$AX_INPUTRAB       ; Initialise update files
```

```
                0000'CF   DD   000F   124          PUSHL   W^SUM$GL_UPDATES
                0000'CF   9F   0013   125          PUSHAB  W^SUM$AX_CBL
00000000'GF       03      FB   0017   126          CALLS   #3,G^SUM$INIT_EDIT
                          001E   127 ;
                  011D    30   001E   128          BSBW    INIT_PAGEHDR                ; Initialize listing page headers
                          0021   129 ;
                          0021   130 ;
                          0021   131 ; This is the main loop.  The next line is read from the input stream
                          0021   132 ; and written to the required output files.
                          0021   133 ;
                          0021   134 10$:
                0000'CF   9F   0021   135          PUSHAB  W^SUM$AX_CBL                ; Get next update line
00000000'GF       01      FB   0025   136          CALLS   #1,G^SUM$LINE
                  20 50   E9   002C   137          BLBC    R0,20$                      ; Error if LBC
                          002F   138 ;
          01    0022'CF   B1   002F   139          CMPW    W^SUM$AX_INPUTRAB+RAB$W_RSZ,#1 ; Was record size 1?
                    0B    12   0034   140          BNEQ    15$                         ; No if NEQ
          0C    0000'CF   91   0036   141          CMPB    W^SUM$AT_BUFFER,#FF         ; Was record a single form-feed?
                    04    12   003B   142          BNEQ    15$                         ; No if NEQ
            00 6B   04    E2   003D   143          BBSS    #SSL_V_FORMFEED,(R11),15$ ; Set form-feed flag
                          0041   144 15$:
                  0150    30   0041   145          BSBW    OUTPUT_LINE                 ; Write line to output file
                 25 50    E9   0044   146          BLBC    R0,40$                      ; Error if LBC
                          0047   147 ;
                  01B1    30   0047   148          BSBW    LIST_LINE                   ; Write line to listing file
                 1F 50    E9   004A   149          BLBC    R0,40$                      ; Error if LBC
                    D2    11   004D   150          BRB     10$
                          004F   151 ;
                          004F   152 ; Error processing
                          004F   153 ;
                          004F   154 20$:
00000000'8F       50      D1   004F   155          CMPL    R0,#RMS$_EOF                ; Is error end-of-file?
                    11    13   0056   156          BEQL    30$                         ; Yes if EQL
                          0058   157 ;
                          0058   158 ; Extract facility code and if SUM error form and print message
                          0058   159 ;
51    50    0C    10      EF   0058   160          EXTZV   #STS$V_FAC_NO, -            ; Get facility code
                          005D   161                  #STS$S_FAC_NO,R0,R1             ; into R1
      0084  8F      51      B1   005D   162          CMPW    R1,#<SUM$_NORMAL@-16>       ; Is error from SUM?
                    08    12   0062   163          BNEQ    40$                         ; No if NEQ
                  02AE    30   0064   164          BSBW    SUM_ERROR                   ; Report error
                    B8    11   0067   165          BRB     10$                         ; and go back for next line
                          0069   166 ;
                          0069   167 ; End-of-file processing
                          0069   168 ;
                          0069   169 30$:
                  0097    30   0069   170          BSBW    CLOSE_FILES                 ; Close all files
                          006C   171 40$:
                    05    006C   172          RSB
```

```
                        006D   174          .SBTTL  OPEN_FILES
                        006D   175 ;
                        006D   176 ;++
                        006D   177 ;
                        006D   178 ; Functional description:
                        006D   179 ;
                        006D   180 ; This routine opens all input files and creates all required output files
                        006D   181 ;
                        006D   182 ;
                        006D   183 ; Calling sequence:
                        006D   184 ;
                        006D   185 ;       BSB     OPEN_FILES
                        006D   186 ;
                        006D   187 ;
                        006D   188 ; Input parameters:
                        006D   189 ;
                        006D   190 ;       None
                        006D   191 ;
                        006D   192 ;
                        006D   193 ; Implicit inputs:
                        006D   194 ;
                        006D   195 ;       SUM$AX_INPUTFAB = Input file FAB
                        006D   196 ;       SUM$AX_INPUTRAB = Input file RAB
                        006D   197 ;       SUM$AX_INPUTNAM = Input file NAM block
                        006D   198 ;       SUM$GL_UPDATES  = Update files list address
                        006D   199 ;       SUM$AX_OUTPUFAB = Output file FAB
                        006D   200 ;       SUM$AX_OUTPURAB = Output file RAB
                        006D   201 ;       SUM$AX_LISTFAB  = List file FAB
                        006D   202 ;       SUM$AX_LISTRAB  = List file RAB
                        006D   203 ;
                        006D   204 ;
                        006D   205 ; Output parameters:
                        006D   206 ;
                        006D   207 ;       None
                        006D   208 ;
                        006D   209 ;
                        006D   210 ; Implicit outputs:
                        006D   211 ;
                        006D   212 ;       None
                        006D   213 ;
                        006D   214 ;
                        006D   215 ; Side effects:
                        006D   216 ;
                        006D   217 ;       None
                        006D   218 ;
                        006D   219 ;--
                        006D   220 ;
                        006D   221 OPEN_FILES:
                        006D   222 ;
                        006D   223 ; Open input file and connect FAB to RAB
                        006D   224 ;
                        006D   225          $OPEN   FAB = W^SUM$AX_INPUTFAB - ; Open input file
                        006D   226                  ERR = G^SUM$OPEN_ERR
41 50     E9            007E   227          BLBC    R0,15$                    ; Error if LBC
                        0081   228 ;
                        0081   229          $CONNECT RAB= W^SUM$AX_INPUTRAB - ; Connenct input file to RAB
                        0081   230                  ERR = G^SUM$OPEN_ERR
```

```
                 6D 50    E9  0092   231          BLBC    R0,30$                  ; Error if LBC
                             0095   232 ;
                             0095   233 ;
                             0095   234 ; If there is an update files list call SUM procedure to
                             0095   235 ; open all update files.  The input file NAM block is used
                             0095   236 ; to supply the initial default values.
                             0095   237 ;
               0000'CF   D5  0095   238          TSTL    W^SUM$GL_UPDATES        ; Any updates?
                    12   13  0099   239          BEQL    10$                     ; No if EQL
               0000'CF   9F  009B   240          PUSHAB  W^SUM$GL_UPDATES        ; Open update files
               0000'CF   9F  009F   241          PUSHAB  W^SUM$AX_INPUTNAM
     00000000'GF    02   FB  00A3   242          CALLS   #2,G^SUM$OPEN
                 55 50    E9  00AA   243          BLBC    R0,30$                  ; Error if LBC
                             00AD   244 ;
                             00AD   245 ; If output file was requested create output file and connect RAB to FAB.
                             00AD   246 ;
                             00AD   247 10$:
              28 6B    00   E1  00AD   248          BBC     #SSL_V_OUTPUT,(R11),20$ ; Branch if no output file
                             00B1   249          $CREATE FAB = W^SUM$AX_OUTPUFAB - ; Create output file
                             00B1   250                  ERR = G^SUM$OPEN_ERR
                 3D 50    E9  00C2   251 15$:     BLBC    R0,30$                  ; Error if LBC
                             00C5   252 ;
                             00C5   253          $CONNECT RAB= W^SUM$AX_OUTPURAB - ; Connect output file to RAB
                             00C5   254                  ERR = G^SUM$OPEN_ERR
                 29 50    E9  00D6   255          BLBC    R0,30$                  ; Error if LBC
                             00D9   256 ;
                             00D9   257 ; If listing file was requested create listing file and connect RAB to FAB.
                             00D9   258 ;
                             00D9   259 20$:
              25 6B    02   E1  00D9   260          BBC     #SSL_V_LIST,(R11),30$   ; Branch if no listing file
                             00DD   261          $CREATE FAB = W^SUM$AX_LISTFAB - ; Create listing file
                             00DD   262                  ERR = G^SUM$OPEN_ERR
                    11 50    E9  00EE   263          BLBC    R0,30$
                             00F1   264          $CONNECT RAB= W^SUM$AX_LISTRAB - ; Connect listing file to RAB
                             00F1   265                  ERR = G^SUM$OPEN_ERR
                             0102   266 30$:
                          05 0102   267          RSB
```

```
                     0103   269          .SBTTL  CLOSE_FILES
                     0103   270 ;
                     0103   271 ;++
                     0103   272 ;
                     0103   273 ; Functional description:
                     0103   274 ;
                     0103   275 ; This routine closes all input and output files.
                     0103   276 ;
                     0103   277 ;
                     0103   278 ; Calling sequence:
                     0103   279 ;
                     0103   280 ;       BSB     CLOSE_FILES
                     0103   281 ;
                     0103   282 ;
                     0103   283 ; Input parameters:
                     0103   284 ;
                     0103   285 ;       None
                     0103   286 ;
                     0103   287 ;
                     0103   288 ; Implicit inputs:
                     0103   289 ;
                     0103   290 ;       SUM$AX_INPUTFAB = Input file FAB
                     0103   291 ;       SUM$AX_OUTPUFAB = Output file FAB
                     0103   292 ;       SUM$AX_LISTFAB  = List file FAB
                     0103   293 ;       SUM$AX_CBL      = SUM control block
                     0103   294 ;
                     0103   295 ;
                     0103   296 ; Output parameters:
                     0103   297 ;
                     0103   298 ;       None
                     0103   299 ;
                     0103   300 ;
                     0103   301 ; Implicit outputs:
                     0103   302 ;
                     0103   303 ;       None
                     0103   304 ;
                     0103   305 ;
                     0103   306 ; Side effects:
                     0103   307 ;
                     0103   308 ;       None
                     0103   309 ;
                     0103   310 ;--
                     0103   311 ;
                     0103   312 CLOSE_FILES:
         0000'CF  9F 0103   313          PUSHAB  W^SUM$AX_CBL            ; Close update files
00000000'GF  01  FB 0107   314          CALLS   #1,G^SUM$CLOSE
                     010E   315 ;
                     010E   316          $CLOSE  FAB = SUM$AX_INPUTFAB
                     011B   317 ;
      0D 6B  00  E1  011B   318          BBC     #SSL_V_OUTPUT,(R11),10$ ; Branch if no output file
                     011F   319          $CLOSE  FAB = SUM$AX_OUTPUFAB
                     012C   320 10$:
      0D 6B  02  E1  012C   321          BBC     #SSL_V_LIST,(R11),20$   ; Branch if no listing file
                     0130   322          $CLOSE  FAB = SUM$AX_LISTFAB
                     013D   323 20$:
             05      013D   324          RSB
```

```
                     013E    326          .SBTTL  INIT_PAGEHDR
                     013E    327 ;
                     013E    328 ;++
                     013E    329 ;
                     013E    330 ; Functional description:
                     013E    331 ;
                     013E    332 ; This routine initialises the control variables and title line used
                     013E    333 ; to format a listing.
                     013E    334 ;
                     013E    335 ;
                     013E    336 ; Calling sequence:
                     013E    337 ;
                     013E    338 ;       BSB     INIT_PAGEHDR
                     013E    339 ;
                     013E    340 ;
                     013E    341 ; Input parameters:
                     013E    342 ;
                     013E    343 ;       None
                     013E    344 ;
                     013E    345 ;
                     013E    346 ; Implicit inputs:
                     013E    347 ;
                     013E    348 ;       None
                     013E    349 ;
                     013E    350 ;
                     013E    351 ; Output parameters:
                     013E    352 ;
                     013E    353 ;       None
                     013E    354 ;
                     013E    355 ;
                     013E    356 ; Implicit outputs:
                     013E    357 ;
                     013E    358 ;       SUM$GW_PAGENO   = Initial page number
                     013E    359 ;       SUM$GW_PAGESZ   = Page size (lines/page)
                     013E    360 ;       SUM$AT_TITLEDT  = Title line current date/time
                     013E    361 ;
                     013E    362 ;
                     013E    363 ; Side effects:
                     013E    364 ;
                     013E    365 ;       None
                     013E    366 ;
                     013E    367 ;--
                     013E    368 ;
                     013E    369 INIT_PAGEHDR:
          51 6B   02 E1      013E    370          BBC     #SSL_V_LIST,(R11),20$ ; Skip this if no listing
 00000000'GF   00 FB      0142    371          CALLS   #0,G^LIB$LP_LINES     ; Get system lines/page
0000'CF   50   09 A3      0149    372          SUBW3   #9,R0,W^SUM$GW_PAGESZ ; Set page size allowing for 3 line top
                           014F    373                                        ; margin, 3 line title/subtitle/blank
                           014F    374                                        ; and 3 line bottom margin
0000'CF   0000'CF B0      014F    375          MOVW    W^SUM$GW_PAGESZ, -    ; Cause initial new page
                           0156    376                  W^SUM$GW_LINENO
          0000'CF B4      0156    377          CLRW    W^SUM$GW_PAGENO       ; Initialise page number
                           015A    378          $ASCTIM_S ,W^SUM$GQ_TITLEDS   ; Get current date/time and put into
                           016B    379                                        ; title line
                           016B    380          $ASCTIM_S ,W^SUM$GQ_SBTTLDS, - ; Get input file creation date/time and
                           016B    381                  W^SUM$AX_INPUTXAB+XAB$Q_CDT ; convert to ASCII
     51   0003'CF 9A      017E    382          MOVZBL  W^SUM$AX_INPUTNAM+NAM$B_RSL,R1 ; Get resultant file-spec length
```

```
                 2D  51   D1  0183   383          CMPL    R1,#45                  ; Is it > 45?
                     03   15  0186   384          BLEQ    10$                     ; No if LEQ
                 51  2D   D0  0188   385          MOVL    #45,R1                  ; Set to maximum size
                                0188   386 10$:
      0000'CF    0000'CF  51   28  0188   387          MOVC3   R1,W^SUM$AT_INPUTFN, -  ; Move file name into subtitle buffer
                                0193   388                  W^SUM$AT_SBTTLFS
                                0193   389 20$:
                                05  0193   390          RSB
```

```
                     0194   392          .SBTTL  OUTPUT_LINE
                     0194   393 ;
                     0194   394 ;++
                     0194   395 ;
                     0194   396 ; Functional description:
                     0194   397 ;
                     0194   398 ; This routine writes the current line to the output file.  If the /HEAD
                     0194   399 ; qualifier was specified the record will have a 12 byte record header buffer.
                     0194   400 ;
                     0194   401 ;
                     0194   402 ; Calling sequence:
                     0194   403 ;
                     0194   404 ;       BSB     OUTPUT_LINE
                     0194   405 ;
                     0194   406 ;
                     0194   407 ; Input parameters:
                     0194   408 ;
                     0194   409 ;       None
                     0194   410 ;
                     0194   411 ;
                     0194   412 ; Implicit inputs:
                     0194   413 ;
                     0194   414 ;       SUM$AX_INPUTRAB = Input file RAB
                     0194   415 ;       SUM$AX_OUTPURAB = Output file RAB
                     0194   416 ;       SUM$GL_FLAGS    = Flags word
                     0194   417 ;       SUM$AX_CBL      = SUM control block
                     0194   418 ;
                     0194   419 ;
                     0194   420 ; Output parameters:
                     0194   421 ;
                     0194   422 ;       None
                     0194   423 ;
                     0194   424 ;
                     0194   425 ; Implicit outputs:
                     0194   426 ;
                     0194   427 ;       SUM$AX_RHB      = Record header buffer
                     0194   428 ;
                     0194   429 ;
                     0194   430 ; Side effects:
                     0194   431 ;
                     0194   432 ;       None
                     0194   433 ;
                     0194   434 ;--
                     0194   435 ;
                     0194   436 OUTPUT_LINE:
          50   01   D0  0194   437          MOVL    #SS$_NORMAL,R0          ; Assume successful completion
                     0197   438 ;
                     0197   439 ; If output file not requested then exit from routine
                     0197   440 ;
        5F 6B   00   E1  0197   441          BBC     #SSL_V_OUTPUT,(R11),30$ ; Branch if no output file
                     019B   442 ;
                     019B   443 ; Get size of input record and use to set size of output record
                     019B   444 ;
0022'CF  0022'CF   B0  019B   445          MOVW    W^SUM$AX_INPUTRAB+RAB$W_RSZ, - ; Copy input record size
                     01A2   446                  W^SUM$AX_OUTPURAB+RAB$W_RSZ    ; to output RAB
                     01A2   447 ;
                     01A2   448 ; If /HEAD was not requested then ready to output line, otherwise
```

```
                41 6B    03    E1  01A2   449          BBC     #SSL_V_HEADER,(R11),20$ ; Branch if no RHB field in output file
                                  01A6   450 ;
                                  01A6   451 ; Fill Record header field with information from SUM control block
                                  01A6   452 ;
                56    0000'CF    9E  01A6   453          MOVAB   W^SUM$AX_CBL,R6         ; Get control block address
                57    0000'CF    9E  01AB   454          MOVAB   W^SUM$AX_RHB,R7         ; and record header address
        67   0C   00    67    00    2C  01B0   455          MOVC5   #0,(R7),#0, -           ; Clear record header buffer
                                  01B6   456                  #SSL$RHBSZE,(R7)
              10 1C A6    02    E0  01B6   457          BBS     #SUM_V_SRCUPD, -        ; Branch if line came from update file
                                  01BB   458                  SUM_B_FLAGS(R6),10$
                                  01BB   459 ;
                                  01BB   460 ; Line came from source file
                                  01BB   461 ;
                   67    18 A6    B0  01BB   462          MOVW    SUM_W_LINE_NO(R6), -    ; Copy line number into record
                                  01BF   463                  SHB_W_LINE_NO(R7)       ; header buffer
              23 1C A6    04    E1  01BF   464          BBC     #SUM_V_DELETE, -        ; Branch if no deleted lines information
                                  01C4   465                  SUM_B_FLAGS(R6),20$     ; with this source line
              02 A7    1A A6    B0  01C4   466          MOVW    SUM_W_INSERT_NO(R6), -  ; Copy number of deleted lines into
                                  01C9   467                  SHB_W_INSERT_NO(R7)     ; record header buffer
                       1C    11  01C9   468          BRB     20$
                                  01CB   469 ;
                                  01CB   470 ; Line came from an update file
                                  01CB   471 ;
                                  01CB   472 10$:
              02 A7    1A A6    B0  01CB   473          MOVW    SUM_W_INSERT_NO(R6), -  ; Copy insert number into record
                                  01D0   474                  SHB_W_INSERT_NO(R7)     ; header buffer
              12 1C A6    00    E1  01D0   475          BBC     #SUM_V_AUDIT, -         ; Branch if audit trail switched off
                                  01D5   476                  SUM_B_FLAGS(R6),20$
                   51    08 A6    3C  01D5   477          MOVZWL  SUM_Q_AUDDS(R6),R1      ; Get audit trail string size
                      08    51    D1  01D9   478          CMPL    R1,#SHB_K_AUDSZ         ; Is greater than maximum size?
                            03    15  01DC   479          BLEQ    15$                     ; No if LEQ
                         51    08    D0  01DE   480          MOVL    #SHB_K_AUDSZ,R1         ; Reduce size to maximum
                                  01E1   481 15$:
        04 A7    0C B6    51    28  01E1   482          MOVC3   R1,@SUM_Q_AUDDS+4(R6), -; Copy audit trail string into
                                  01E7   483                  SHB_T_AUDIT(R7)         ; record header buffer
                                  01E7   484 ;
                                  01E7   485 ; Put line (with record header buffer) to output file
                                  01E7   486 ;
                                  01E7   487 20$:
                                  01E7   488          $PUT    RAB = SUM$AX_OUTPURAB -
                                  01E7   489                  ERR = G^SUM$WRITE_ERR
                                  01FA   490 30$:
                         05       01FA   491          RSB
```

```
                                        .SBTTL  LIST_LINE
                                ;
                                ;++
                                ;
                                ; Functional description:
                                ;
                                ; This routine forms a listing line which shows the audit trail, source
                                ; line number or insert line number and the source or update line.
                                ;
                                ;
                                ; Calling sequence:
                                ;
                                ;       BSB     LIST_LINE
                                ;
                                ;
                                ; Input paramters:
                                ;
                                ;       None
                                ;
                                ;
                                ; Implicit inputs:
                                ;
                                ;       SUM$AX_CBL      = SUM control block
                                ;       SUM_GL_FLAGS    = Flags wurd
                                ;       SUM$AT_AUDIT    = Listing line buffer
                                ;
                                ;
                                ; Output paramters:
                                ;
                                ;       None
                                ;
                                ;
                                ; Implicit outputs:
                                ;
                                ;       None
                                ;
                                ;
                                ; Side effects:
                                ;
                                ;       None
                                ;
                                ;--
                                ;
                                LIST_LINE:
                        50  01  D0  01FB  537         MOVL    #SS$_NORMAL,R0          ; Assume successful completion
                      66 6B 02  E1  01FE  538         BBC     #SSL_V_LIST,(R11),60$   ; Branch if no listing file
                                0202  539 ;
                                0202  540 ; Clear audit trail and line number fields
                                0202  541 ;
                    57  0000'CF  9E  0202  542         MOVAB   W^SUM$AT_AUDIT,R7       ; Set pointer to audit/line no. fields
67  18  20  67  00  2C  0207  543         MOVC5   #0,(R7),#^A/ /, -       ; Fill these fields with spaces
                                020D  544                 #SSL$AULSZE,(R7)
                                020D  545 ;
                    56  0000'CF  9E  020D  546         MOVAB   W^SUM$AX_CBL,R6         ; Get SUM control block address
                  1D 1C A6  02  E0  0212  547         BBS     #SUM_V_SRCUPD, -        ; Branch if updated line
                                0217  548                 SUM_B_FLAGS(R6),20$
                                0217  549 ;
```

```
                                0217   550 ; Line has come from source file.
                                0217   551 ; If there is deleted line information put '-n' into audit trail field,
                                0217   552 ; and put source line number into line number field
                                0217   553 ;
        0A 1C A6    04    E1    0217   554         BBC     #SUM_V_DELETE, -        ; Branch if no deleted lines information
                                021C   555                 SUM_B_FLAGS(R6),10$
              87    2D    90    021C   556         MOVB    #^A7-7,(R7)+            ; Insert leading '-' character
           52   1A A6    3C    021F   557         MOVZWL  SUM_W_INSERT_NO(R6),R2  ; Get number of deleted lines
                 00CA    30    0223   558         BSBW    DEC_LTOR                ; and output into audit trail field
                                0226   559 10$:
           52   18 A6    3C    0226   560         MOVZWL  SUM_W_LINE_NO(R6),R2    ; Get source line number
        57 0000'CF    9E    022A   561         MOVAB   W^SOM$AT_LINENE,R7      ; Set address to put number
                 00D3    30    022F   562         BSBW    DEC_RTOL                ; Output number into line number field
                   1A    11    0232   563         BRB     40$
                                0234   564 ;
                                0234   565 ; Line has come from an update file.
                                0234   566 ; If audit trailing is switched on copy audit trail into audit trail field,
                                0234   567 ; and put insert line number into line number field.
                                0234   568 ;
                                0234   569 20$:
        06 1C A6    00    E1    0234   570         BBC     #SUM_V_AUDIT, -         ; Branch if audit trailing switched off
                                0239   571                 SUM_B_FLAGS(R6),30$
     67 0C B6    08 A6    28    0239   572         MOVC3   SUM_Q_AUDDS(R6), -      ; Copy audit trail string into
                                023F   573                 @SUM_Q_AUDDS+4(R6),(R7) ; audit trail field
                                023F   574 30$:
           52   1A A6    3C    023F   575         MOVZWL  SUM_W_INSERT_NO(R6),R2  ; Get insert line number
        57 0006'CF    9E    0243   576         MOVAB   W^SOM$AT_LINENO+6,R7    ; Set field address
                 00BA    30    0248   577         BSBW    DEC_RTOL                ; Output number into field
              77   2E    90    024B   578         MOVB    #^A7./,-(R7)            ; and add leading '.' character
                                024E   579 ;
                                024E   580 ;
                                024E   581 ; Add size of audit trail/line number fields to size of line and if
                                024E   582 ; over 132 characters reduce to 132 characters, then print line
                                024E   583 ;
                                024E   584 40$:
0022'CF    18    A1    024E   585         ADDW3   #SSL$AULSZE, -
              58          0253   586                 W^SUM$AX_INPUTRAB+RAB$W_RSZ,R8
00FF 8F    58    B1    0254   587         CMPW    R8,#SSL$BUFSZE          ; Is total size > SSL$BUFSZE
           05    15    0259   588         BLEQ    50$                     ; No if LEQ
58   00FF 8F    B0    025B   589         MOVW    #SSL$BUFSZE,R8          ; Set size to SSL$BUFSZE bytes
                                0260   590 50$:
59   0000'CF    9E    0260   591         MOVAB   W^SUM$AT_AUDIT,R9       ; Set address of line
        0001    30    0265   592         BSBW    PRINT_LINE              ; Print line
                                0268   593 60$:
                    05  0268   594         RSB
```

```
                     0269   596          .SBTTL  PRINT_LINE
                     0269   597 ;
                     0269   598 ;++
                     0269   599 ;
                     0269   600 ; Functional description:
                     0269   601 ;
                     0269   602 ; This routine outputs a single line to the listing file.  If the current
                     0269   603 ; page is full the line is printed on a new page.
                     0269   604 ;
                     0269   605 ;
                     0269   606 ; Calling sequence:
                     0269   607 ;
                     0269   608 ;       BSB     PRINT_LINE
                     0269   609 ;
                     0269   610 ;
                     0269   611 ; Input parameters:
                     0269   612 ;
                     0269   613 ;       R8 = Record size
                     0269   614 ;       R9 = Record address
                     0269   615 ;
                     0269   616 ;
                     0269   617 ; Implicit inputs:
                     0269   618 ;
                     0269   619 ;       SUM$GL_FLAGS    = Flags word
                     0269   620 ;
                     0269   621 ;
                     0269   622 ; Output parameters:
                     0269   623 ;
                     0269   624 ;       None
                     0269   625 ;
                     0269   626 ;
                     0269   627 ; Implicit outputs:
                     0269   628 ;
                     0269   629 ;       None
                     0269   630 ;
                     0269   631 ;
                     0269   632 ; Side effects:
                     0269   633 ;
                     0269   634 ;       None
                     0269   635 ;
                     0269   636 ;--
                     0269   637 ;
                     0269   638 PRINT_LINE:
          50    01   D0  0269   639          MOVL    #SS$_NORMAL,R0          ; Assume successful completion
        63 6B    02   E1  026C   640          BBC     #SSL_V_LIST,(R11),30$   ; Branch if no listing file
0000'CF   0000'CF   B1  0270   641          CMPW    W^SUM$GW_LINENO,W^SUM$GW_PAGESZ ; Is current page full?
             45   19  0277   642          BLSS    20$                     ; No if LSS
          0300 8F   BB  0279   643          PUSHR   #^M<R8,R9>              ; Save line size/address
          0000'CF   B6  027D   644          INCW    W^SUM$GW_PAGENO         ; Increment page number
      52  0000'CF   3C  0281   645          MOVZWL  W^SUM$GW_PAGENO,R2      ; Get current page number
      57  0000'CF   DE  0286   646          MOVAL   W^SUM$AT_TITLEPN,R7     ; and address in title line
             0077   30  028B   647          BSBW    DEC_RTOL                ; Convert to ASCII
      58  0000'8F   3C  028E   648          MOVZWL  #SUM$K_TITLELN,R8       ; Get title line size
      59  0000'CF   DE  0293   649          MOVAL   W^SUM$AT_TITLE,R9       ; and address
             0039   30  0298   650          BSBW    PUT_LINE                ; and print
           15 50    E9  029B   651          BLBC    R0,TO$                  ; Error if LBC
      58  0000'8F   3C  029E   652          MOVZWL  #SUM$K_SBTTLLN,R8       ; Get subtitle line size
```

```
                59   0000'CF   DE   02A3   653          MOVAL   W^SUM$AT_SBTTL,R9       ; and address
                       0029    30   02A8   654          BSBW    PUT_LINE               ; and print
                     05 50     E9   02AB   655          BLBC    R0,10$                 ; Error if LBC
                        58     D4   02AE   656          CLRL    R8                     ; Print blank line
                       0021    30   02B0   657          BSBW    PUT_LINE
                                   02B3   658 10$:
                   0000'CF     B4   02B3   659          CLRW    W^SUM$GW_LINENO        ; Reset line number
                   0300 8F     BA   02B7   660          POPR    #^M<R8,R9>             ; Restore line size/address
                     15 50     E9   02BB   661          BLBC    R0,30$                 ; Error if LBC
                                   02BE   662 20$:
                   0000'CF     B6   02BE   663          INCW    W^SUM$GW_LINENO        ; Increment line number
                      000F     30   02C2   664          BSBW    PUT_LINE               ; Print line
                     0B 50     E9   02C5   665          BLBC    R0,30$                 ; Error if LBC
                07 6B    04    E5   02C8   666          BBCC    #SSL_V_FORMFEED,(R11),30$ ; Branch if not FF record
0000'CF            0000'CF     B0   02CC   667          MOVW    W^SUM$GW_PAGESZ, -     ; Make page full
                                   02D3   668                  W^SUM$GW_LINENO
                                   02D3   669 30$:
                               05  02D3   670          RSB
```

```
                        02D4    672          .SBTTL  PUT_LINE
                        02D4    673 ;++
                        02D4    674 ;
                        02D4    675 ; Functional description:
                        02D4    676 ;
                        02D4    677 ; This routine puts a single record to the listing file.
                        02D4    678 ;
                        02D4    679 ;
                        02D4    680 ; Calling sequence:
                        02D4    681 ;
                        02D4    682 ;       BSB     PUT_LINE
                        02D4    683 ;
                        02D4    684 ;
                        02D4    685 ; Input parameters:
                        02D4    686 ;
                        02D4    687 ;       R8 = Record size
                        02D4    688 ;       R9 = Record address
                        02D4    689 ;
                        02D4    690 ;
                        02D4    691 ; Implicit inputs:
                        02D4    692 ;
                        02D4    693 ;       SUM$AX_LISTRAB  = Listing file RAB
                        02D4    694 ;
                        02D4    695 ;
                        02D4    696 ; Output parameters:
                        02D4    697 ;
                        02D4    698 ;       None
                        02D4    699 ;
                        02D4    700 ;
                        02D4    701 ; Implicit outputs:
                        02D4    702 ;
                        02D4    703 ;       None
                        02D4    704 ;
                        02D4    705 ;
                        02D4    706 ; Side effects:
                        02D4    707 ;
                        02D4    708 ;       None
                        02D4    709 ;
                        02D4    710 PUT_LINE:
0022'CF    58    B0     02D4    711          MOVW    R8,W^SUM$AX_LISTRAB+RAB$W_RSZ   ; Set record size
0028'CF    59    D0     02D9    712          MOVL    R9,W^SUM$AX_LISTRAB+RAB$L_RBF   ; and buffer address
                        02DE    713          $PUT    RAB=W^SUM$AX_LISTRAB, -
                        02DE    714                  ERR=G^SUM$WRITE_ERR
                   05   02EF    715          RSB
```

```
                                02F0   717          .SBTTL  DEC_LTOR
                                02F0   718 ;
                                02F0   719 ; ++
                                02F0   720 ;
                                02F0   721 ; Functional descrition:
                                02F0   722 ;
                                02F0   723 ;       This routine outputs a decimal number left to right
                                02F0   724 ;
                                02F0   725 ;
                                02F0   726 ; Calling sequence:
                                02F0   727 ;
                                02F0   728 ;       BSB     DEC_LTOR
                                02F0   729 ;
                                02F0   730 ;
                                02F0   731 ; Input paramters:
                                02F0   732 ;
                                02F0   733 ;       R2 = Number to output
                                02F0   734 ;       R7 = Address to put output characters
                                02F0   735 ;
                                02F0   736 ;
                                02F0   737 ; Implicit inputs:
                                02F0   738 ;
                                02F0   739 ;       None
                                02F0   740 ;
                                02F0   741 ;
                                02F0   742 ; Output parameters:
                                02F0   743 ;
                                02F0   744 ;       R7 = Address of byte after last digit of number
                                02F0   745 ;
                                02F0   746 ;
                                02F0   747 ; Implicit outputs:
                                02F0   748 ;
                                02F0   749 ;       None
                                02F0   750 ;
                                02F0   751 ;
                                02F0   752 ; Side effects:
                                02F0   753 ;
                                02F0   754 ;       R2, R3 zeroed
                                02F0   755 ;
                                02F0   756 ;--
                                02F0   757 ;
                                02F0   758 DEC_LTOR:
                          53    D4    02F0   759          CLRL    R3                      ; Clear high register
                                02F2   760 10$:
50      52      52        0A    78    02F2   761          EDIV    #10,R2,R2,R0            ; Get least significant digit
        7E      50        30    81    02F7   762          ADDB3   #^A/0/,R0,-(SP)         ; Convert to ASCII char and stack
                          52    D5    02FB   763          TSTL    R2                      ; All of number converted?
                          02    13    02FD   764          BEQL    20$                     ; Yes if EQL
                          F1    10    02FF   765          BSBB    10$             ; Call routine again to convert next char
                                0301   766 20$:
                    87    8E    90    0301   767          MOVB    (SP)+,(R7)+             ; Move character to output buffer
                                05    0304   768          RSB
```

```
                     0305   770          .SBTTL  DEC_RTOL
                     0305   771 ;
                     0305   772 ; ++
                     0305   773 ;
                     0305   774 ; Functional descrition:
                     0305   775 ;
                     0305   776 ;       This routine outputs a decimal number right to left
                     0305   777 ;
                     0305   778 ;
                     0305   779 ; Calling sequence:
                     0305   780 ;
                     0305   781 ;       BSB     DEC_RTOL
                     0305   782 ;
                     0305   783 ;
                     0305   784 ; Input paramters:
                     0305   785 ;
                     0305   786 ;       R2 = Number to output
                     0305   787 ;       R7 = Address immediately after right margin of number
                     0305   788 ;
                     0305   789 ;
                     0305   790 ; Implicit inputs:
                     0305   791 ;
                     0305   792 ;       None
                     0305   793 ;
                     0305   794 ;
                     0305   795 ; Output parameters:
                     0305   796 ;
                     0305   797 ;       R7 = Address of most significant digit of number
                     0305   798 ;
                     0305   799 ;
                     0305   800 ; Implicit outputs:
                     0305   801 ;
                     0305   802 ;       None
                     0305   803 ;
                     0305   804 ;
                     0305   805 ; Side effects:
                     0305   806 ;
                     0305   807 ;       R2, R3 zeroed
                     0305   808 ;
                     0305   809 ;--
                     0305   810 ;
                     0305   811 DEC_RTOL:
            53   D4  0305   812          CLRL    R3                      ; Clear high register
                     0307   813 10$:
50  52   52  0A   7B  0307   814          EDIV    #10,R2,R2,R0            ; Get least significant digit
    77   50  30   81  C30C   815          ADDB3   #^A/0/,R0,-(R7)         ; Convert to ASCII char and put in
                     0310   816                                          ; output buffer
            52   D5  0310   817          TSTL    R2                      ; All of number converted?
            F3   12  0312   818          BNEQ    10$                     ; No if NEQ
                 05  0314   819          RSB
```

```
                     0315   821          .SBTTL  SUM_ERROR
                     0315   822 ;
                     0315   823 ;++
                     0315   824 ;
                     0315   825 ; Functional description:
                     0315   826 ;
                     0315   827 ;       This routine forms and prints error messages which describe
                     0315   828 ;       Source Update Merge errors.  The messages will be output to
                     0315   829 ;       the system error device and the listing file.
                     0315   830 ;
                     0315   831 ;
                     0315   832 ; Calling sequence:
                     0315   833 ;
                     0315   834 ;       BSB     SUM_ERROR
                     0315   835 ;
                     0315   836 ;
                     0315   837 ; Input parameters:
                     0315   838 ;
                     0315   839 ;       R0 = Error code
                     0315   840 ;
                     0315   841 ;
                     0315   842 ; Implicit inputs:
                     0315   843 ;
                     0315   844 ;       SUM$AX_CBL is SUM control block
                     0315   845 ;
                     0315   846 ;
                     0315   847 ; Output parameters:
                     0315   848 ;
                     0315   849 ;       None
                     0315   850 ;
                     0315   851 ;
                     0315   852 ; Implicit outputs:
                     0315   853 ;
                     0315   854 ;       None
                     0315   855 ;
                     0315   856 ;
                     0315   857 ; Side effects:
                     0315   858 ;
                     0315   859 ;       R0 to R5 is destroyed
                     0315   860 ;
                     0315   861 ;--
                     0315   862 ;
                     0315   863 SUM_ERROR:
    56  0000'CF   9E   0315   864          MOVAB   W^SUM$AX_CBL,R6         ; Get address of SUM control block
00848800 8F   50   D1   031A   865          CMPL    R0,#SUM$_EDITSCLSH      ; Is it 'Edits clash' error?
          05   12   0321   866          BNEQ    10$                     ; No if NEQ
 2E 1C A6   03   E0   0323   867          BBS     #SUM_V_SUBCLSH, -       ; Branch if not first error line
                     0328   868                  SUM_B_FLAGS(R6),30$     ; of error report
                     0328   869 ;
                     0328   870 ; If first error print error message
                     0328   871 ;
                     0328   872 10$:
                     0328   873          $GETMSG_S R0,W^SUM$AW_MSGLEN,-  ; Get error message
                     0328   874                  W^SUM$AQ_MSGDES
      16 50   E9   033D   875          BLBC    R0,30$                  ; Error if LBC
58  0000'CF   3C   0340   876          MOVZWL  W^SUM$AW_MSGLEN,R8      ; Get message length
59  0000'CF   DE   0345   877          MOVAL   W^SUM$AT_MSGBUF,R9      ; and address
```

```
                0050   30  034A   878          BSBW    TERM_LINE                       ; Output to terminal
              4C 50    E9  034D   879          BLBC    R0,90$                          ; Error if LBC
               FF16    30  0350   880          BSBW    PRINT_LINE                      ; Output to listing file
              46 50    E9  0353   881          BLBC    R0,90$                          ; Error if LBC
                       0356   882 ;
                       0356   883 ; Now print erroneous source line
                       0356   884 ;
                       0356   885 30$:
        58   0022'CF    3C  0356   886          MOVZWL  W^SUM$AX_INPUTRAB+RAB$W_RSZ,R8  ; Get line size
        59   0028'CF    D0  035B   887          MOVL    W^SUM$AX_INPUTRAB+RAB$L_RBF,R9  ; and address
                       0360   888 ;
                       0360   889 ; Append name of file from which line came
                       0360   890 ;
        51   59   58    C1  0360   891          ADDL3   R8,R9,R1                        ; Set pointer to end of line
          50   10 A6    3C  0364   892          MOVZWL  SUM_Q_FILESP(R6),R0             ; Get size of file-spec
               58   04   C0  0368   893          ADDL2   #4,R8                           ; Include size of separating spaces
               58   50   C0  036B   894          ADDL2   R0,R8                           ; Compute total size of line
    000000FF   8F   58   D1  036E   895          CMPL    R8,#SSL$BUFSZE                  ; Will it fit in buffer?
                    10   15  0375   896          BLEQ    40$                             ; Yes if LEQ
52  58    000000FF   8F   C3  0377   897          SUBL3   #SSL$BUFSZE,R8,R2               ; Compute size of overflow
               51   52   C2  037F   898          SUBL2   R2,R1                           ; and move pointer back this amount
        58   00FF   8F   3C  0382   899          MOVZWL  #SSL$BUFSZE,R8                  ; Set line size to maximum
                       0387   900 40$:
    81    20202020   8F   D0  0387   901          MOVL    #^A/    /,(R1)+                 ; Insert four spaces
     61    14 B6    50   28  038E   902          MOVC3   R0,@SUM_Q_FILESP+4(R6),(R1)     ; Copy file name to line
                0007   30  0393   903          BSBW    TERM_LINE                       ; Output to terminal
              03 50    E9  0396   904          BLBC    R0,90$                          ; Error if LBC
               FECD    30  0399   905          BSBW    PRINT_LINE                      ; Output to listing file
                       039C   906 ;
                       039C   907 90$:
                    05  039C   908          RSB
```

```
                     039D   910 .SBTTL  TERM_LINE
                     039D   911 ;
                     039D   912 ;
                     039D   913 TERM_LINE:
0022'CF    58    B0  039D   914         MOVW    R8,W^SUM$AX_TERMRAB+RAB$W_RSZ   ; Set record size
0028'CF    59    D0  03A2   915         MOVL    R9,W^SUM$AX_TERMRAB+RAB$L_RBF   ; and buffer address
                     03A7   916         $PUT    RAB = W^SUM$AX_TERMRAB, -
                     03A7   917                 ERR = G^SUM$WRITE_ERR
                  05 03B8   918         RSB
                     03B9   919 ;
                     03B9   920 ;
                     03B9   921         .END
```

```
$$.TMP1                         =  00000002
$$.TMP2                         =  000000CF
BIT...                          =  00000005
CLOSE_FILES                        00000103  R       02
DEC_LTOR                           000002F0  R       02
DEC_RTOL                           00000305  R       02
FF                              =  0000000C
INIT_PAGEHDR                       0000013E  R       02
LIB$CP_LINES                       ********    X     02
LIST_LINE                          000001FB  R       02
NAM$B_RSL                       =  00000003
OPEN_FILES                         0000006D  R       02
OUTPUT_LINE                        00000194  R       02
PRINT_LINE                         00000269  R       02
PUT_LINE                           000002D4  R       02
RAB$L_RBF                       =  00000028
RAB$W_RSZ                       =  00000022
RMS$_EOF                           ********    X     02
SHB_K_AUDSZ                     =  00000008
SHB_K_BLN                          0000000C
SHB_T_AUDIT                        00000004
SHB_W_INSERT_NO                    00000002
SHB_W_LINE_NO                      00000000
SIZ...                          =  00000001
SS$_NORMAL                      =  00000001
SSL$AULSZE                      =  00000018
SSL$AUTSZE                      =  00000010
SSL$BUFSZE                      =  000000FF
SSL$FILSZE                      =  00000040
SSL$LNOSZE                      =  00000006
SSL$RHBSZE                      =  0000000C
SSL_M_FORMFEED                  =  00000010
SSL_M_HEADER                    =  00000008
SSL_M_LIST                      =  00000004
SSL_M_MODE                      =  00000002
SSL_M_OUTPUT                    =  00000001
SSL_V_FORMFEED                  =  00000004
SSL_V_HEADER                    =  00000003
SSL_V_LIST                      =  00000002
SSL_V_MODE                      =  00000001
SSL_V_OUTPUT                    =  00000000
STS$S_FAC_NO                    =  0000000C
STS$V_FAC_NO                    =  00000010
SUM$AQ_MSGDES                      ********    X     02
SUM$AT_AUDIT                       ********    X     02
SUM$AT_BUFFER                      ********    X     02
SUM$AT_INPUTFN                     ********    X     02
SUM$AT_LINENE                      ********    X     02
SUM$AT_LINENO                      ********    X     02
SUM$AT_MSGBUF                      ********    X     02
SUM$AT_SBTTL                       ********    X     02
SUM$AT_SBTTLFS                     ********    X     02
SUM$AT_TITLE                       ********    X     02
SUM$AT_TITLEPN                     ********    X     02
SUM$AW_MSGLEN                      ********    X     02
SUM$AX_CBL                         ********    X     02
SUM$AX_INPUTFAB                    ********    X     02
SUM$AX_INPUTNAM                    ********    X     02
SUM$AX_INPUTRAB                    ********    X     02
SUM$AX_INPUTXAB                    ********    X     02
SUM$AX_LISTFAB                     ********    X     02
SUM$AX_LISTRAB                     ********    X     02
SUM$AX_OUTPUFAB                    ********    X     02
SUM$AX_OUTPURAB                    ********    X     02
SUM$AX_RHB                         ********    X     02
SUM$AX_TERMRAB                     ********    X     02
SUM$CLOSE                          ********    X     02
SUM$GL_FLAGS                       ********    X     02
SUM$GL_UPDATES                     ********    X     02
SUM$GQ_SBTTLDS                     ********    X     02
SUM$GQ_TITLEDS                     ********    X     02
SUM$GW_LINENO                      ********    X     02
SUM$GW_PAGENO                      ********    X     02
SUM$GW_PAGESZ                      ********    X     02
SUM$INIT_EDIT                      ********    X     02
SUM$K_SBTTLLN                      ********    X     02
SUM$K_TITLELN                      ********    X     02
SUM$LINE                           ********    X     02
SUM$OPEN                           ********    X     02
SUM$OPEN_ERR                       ********    X     02
SUM$UPDATE                         00000000  RG      02
SUM$WRITE_ERR                      ********    X     02
SUM$_EDITSCLSH                  =  00848800
SUM$_NORMAL                     =  00848001
SUM_B_FLAGS                        0000001C
SUM_ERROR                          00000315  R       02
SUM_K_BLN                          0000001D
SUM_L_ISDATA                       00000004
SUM_L_STS                          00000000
SUM_M_AUDIT                     =  00000001
SUM_M_AUDITNEW                  =  00000002
SUM_M_DELETE                    =  00000010
SUM_M_SRCUPD                    =  00000004
SUM_M_SUBCLSH                   =  00000008
SUM_Q_AUDDS                        00000008
SUM_Q_FILESP                       00000010
SUM_V_AUDIT                     =  00000000
SUM_V_AUDITNEW                  =  00000001
SUM_V_DELETE                    =  00000004
SUM_V_SRCUPD                    =  00000002
SUM_V_SUBCLSH                   =  00000003
SUM_W_INSERT_NO                    0000001A
SUM_W_LINE_NO                      00000018
SYS$ASCTIM                         ********   GX     02
SYS$CLOSE                          ********   GX     02
SYS$CONNECT                        ********   GX     02
SYS$CREATE                         ********   GX     02
SYS$GETMSG                         ********   GX     02
SYS$OPEN                           ********   GX     02
SYS$PUT                            ********   GX     02
TERM_LINE                          0000039D  R       02
XAB$Q_CDT                          ********    X     02
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
ABS	0000001D (29.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
SUM$CODE	000003B9 (953.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	33	00:00:00.10	00:00:00.56
Command processing	113	00:00:00.62	00:00:03.13
Pass 1	334	00:00:11.29	00:00:26.91
Symbol table sort	0	00:00:01.41	00:00:02.96
Pass 2	156	00:00:02.98	00:00:06.68
Symbol table output	14	00:00:00.11	00:00:00.12
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	654	00:00:16.54	00:00:40.39

The working set limit was 1350 pages.
63551 bytes (125 pages) of virtual memory were used to buffer the intermediate code.
There were 60 pages of symbol table space allocated to hold 1006 non-local and 33 local symbols.
921 source lines were read in Pass 1, producing 19 object records in Pass 2.
31 pages of virtual memory were used to define 29 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SUM.OBJ]SUM.MLB;1	6
_255DUA28:[SYSLIB]STARLET.MLB;2	18
TOTALS (all libraries)	24

1139 GETS were required to define 24 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS=LIS$:UPDATE/OBJ=OBJ$:UPDATE MSRC$:UPDATE/UPDATE=(ENH$:UPDATE)+LIB$:SUM/LIB
```

SUMLIST
LIS

SUMMAIN
LIS

SUMTFRVEC
LIS

SUMFILES
LIS

SUMMSG
LIS

SUMERROR
LIS

SUMOPEN
LIS

SYS

UPDATE
LIS

SYS
MAP