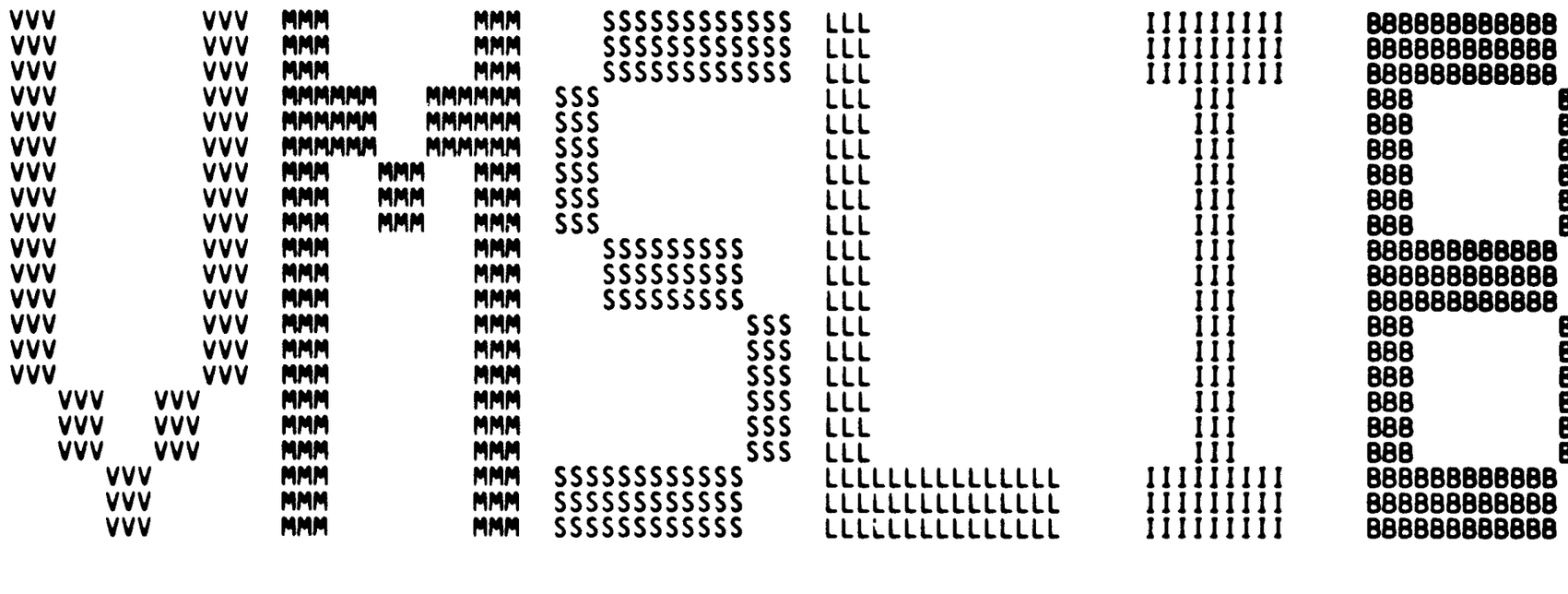
VVV VVV MMM MMM SSSSSSSSSSS LLL IIIIIIIII BBBBBBBBBBB
VVV VVV MMM MMM SSSSSSSSSSS LLL IIIIIIIII BBBBBBBBBBB
VVV VVV MMM MMM SSSSSSSSSSS LLL IIIIIIIII BBBBBBBBBBB
VVV VVV MMMMMM MMMMMM SSS LLL III BBB BBB
VVV VVV MMMMMM MMMMMM SSS LLL III BBB BBB
VVV VVV MMMMMM MMMMMM SSS LLL III BBB BBB
VVV VVV MMM MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSSSSSSSS LLL III BBBBBBBBBBB
VVV VVV MMM MMM SSSSSSSSS LLL III BBBBBBBBBBB
VVV VVV MMM MMM SSSSSSSSS LLL III BBBBBBBBBBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSSSSSSSSSS LLLLLLLLLLLLLLL IIIIIIIII BBBBBBBBBBB
VVV VVV MMM MMM SSSSSSSSSSS LLLLLLLLLLLLLLL IIIIIIIII BBBBBBBBBBB
VVV MMM MMM SSSSSSSSSSS LLLLLLLLLLLLLLL IIIIIIIII BBBBBBBBBBB

FILEID**LIBASSIGN

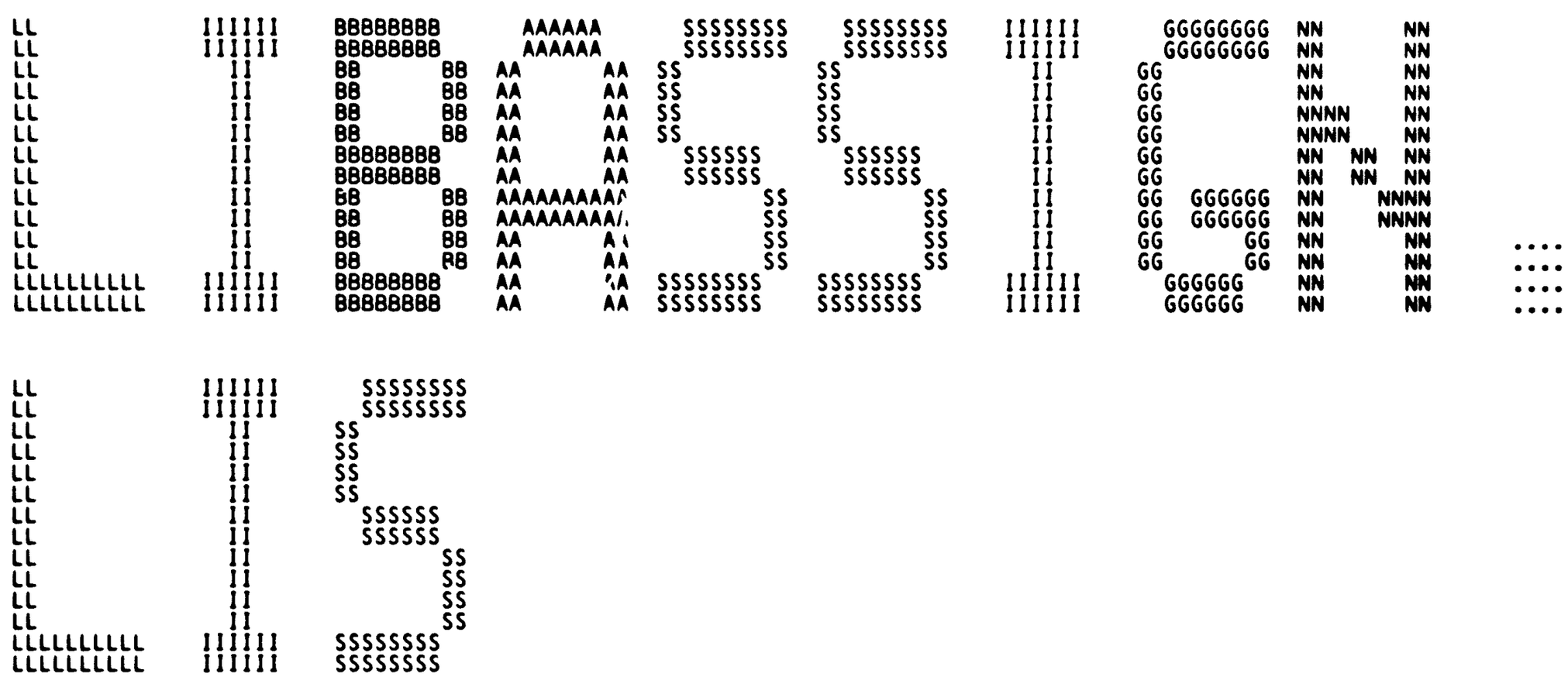

Table of contents

```
0000     1          .TITLE  LIB$ASSIGN              ; Assign channel to device
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 SUPPLIED BY DIGITAL.                  *
0000    23  ;*                                                                           *
0000    24  ;*                                                                           *
0000    25  ;*****************************************************************************
0000    26  ;
0000    27  ;
0000    28  ; FACILITY: Language-independent support library
0000    29  ;++
0000    30  ; ABSTRACT:
0000    31  ;
0000    32  ;
0000    33  ;--
0000    34  ;
0000    35  ; VERSION: 1
0000    36  ;
0000    37  ; HISTORY:
0000    38  ;
0000    39  ; AUTHOR:
0000    40  ;
0000    41  ;       Herb Jacobs             August 1981
0000    42  ;
```

```
0000     44          .SBTTL  HISTORY         ; Detailed Current Edit History
0000     45
0000     46 ; EDIT HISTORY:
0000     47 ;
0000     48 ; 1-001 -
```

```
                        .SBTTL  DECLARATIONS

;
; INCLUDE FILES:
;
                        $FABDEF
                        $NAMDEF
;
; EXTERNAL SYMBOLS:
;

;
; MACROS:
;

;
; PSECT DECLARATIONS:
;

00000000                .PSECT  _LIB$CODE       PIC, SHR, LONG, EXE, NOWRT

;
; EQUATED SYMBOLS:
;
```

```
                                76          .SBTTL  LIB$ASSIGN -
                                77
                                78 ;++
                                79 ; FUNCTIONAL DESCRIPTION:
                                80 ;
                                81 ;       These routines provide a general mechanism to get a channel assigned
                                82 ;       to the associated device with a file specification.  The channel
                                83 ;       assigned is always to the device, never to the file.  The purpose
                                84 ;       of these routines are to successfully accomplish up cased recursive
                                85 ;       logical name translation and succeed with process permanent files.
                                86 ;       Since these routines use RMS as a basis, compatibility is assured
                                87 ;       with all working programs.
                                88 ;
                                89 ; CALLING SEQUENCE:
                                90 ;
                                91 ;       ret = LIB$ASSIGN        ( filedesc, chan)
                                92 ;       ret = LIB$ASSIGN_FILE   ( filedesc, chan)
                                93 ;       ret = LIB$ASSIGN_DEV    ( filedesc, chan)
                                94 ;
                                95 ; INPUT PARAMETERS:
                                96 ;
                                97 ;       FILEDESC - Address of character string descriptor describing input
                                98 ;
                                99 ; IMPLICIT INPUTS:
                               100 ;
                               101 ;       NONE
                               102 ;
                               103 ; OUTPUT PARAMETERS:
                               104 ;
                               105 ;       CHAN -  Address of word to recieve channel number assigned
                               106 ;
                               107 ; IMPLICIT OUTPUTS:
                               108 ;
                               109 ;       NONE
                               110 ;
                               111 ; COMPLETION CODES:
                               112 ;
                               113 ;       SS$_NORMAL      - success
                               114 ;       All other completion codes are generated as RMS or SS codes.
                               115 ;
                               116 ; SIDE EFFECTS:
                               117 ;
                               118 ;       NONE
                               119 ;
                               120 ;--
                               121         .ENTRY  -
                    0000       122 LIB$ASSIGN,     ^M<>
         41'AF    6C   FA      123         CALLG   (AP),B^LIB$ASSIGN_DEV           ;ATTEMPT DEVICE ASSIGNMENT
               16 50   E8      124         BLBS    R0,10$                          ;BRANCH IF SUCCESSFUL
                               125 ;
                               126 ; PROCESS LIST OF RECOGNIZABLE FAILURES AS A DEVICE NAME THAT SHOULD BE
                               127 ; RETURNED AS AN ERROR RATHER THAN ATTEMPTING AS A FILE NAME
                               128 ;
50   00000000'8F    D1          129         CMPL    #RMS$_CHN,R0                    ;DEVICE ALLOCATED
                 0D    13      130         BEQL    10$
50   00000000'8F    D1          131         CMPL    #RMS$_PRV,R0                    ;NO PRIVILEGE FOR DEVICE ACCESS
                 04    13      132         BEQL    10$
```

```
                20'AF    6C    FA   001B   133          CALLG   (AP),B^LIB$ASSIGN_FILE          ;ATTEMPT AS GENERAL FILE SPEC
                             04   001F   134 10$:      RET                                     ;RETURN STATUS FROM SUBROUTINE
                                  0020   135
                                  0020   136
                                  0020   137 ;
                                  0020   138 ; GET CHANNEL ASSOCIATED WITH FILE SPEC, ASSUME NO SPECIAL PUNCTUATION
                                  0020   139 ;
                                  0020   140          .ENTRY  -
                         007C     0020   141 LIB$ASSIGN_FILE,^M<R2,R3,R4,R5,R6>
          5E   000000B0 8F   C2   0022   142          SUBL    #FAB$C_BLN+NAM$C_BLN,SP         ;ALLOCATE ROOM FOR FAB AND NAM
                    56   5E   D0   0029   143          MOVL    SP,R6                           ;GET ADDRESS OF FAB
                         007C   30   002C   144          BSBW    ASSIGN_SETUP                    ;PERFORM COMMON SETUP
                                  002F   145          $PARSE  FAB=<(R6)>                      ;GET THE DEVICE INFORMATION
                      05 50      E9   0038   146 10$:      BLBC    R0,20$                  ;BRANCH IF ERROR TO RETURN CODE
                00F1'CF    6C    FA   003B   147          CALLG   (AP),W^ASSIGN_CHAN              ;CALL FOR STACK CLEANUP
                             04   0040   148 20$:      RET                                     ;CHANNEL
                                  0041   149
                                  0041   150 ;
                                  0041   151 ; GET CHANNEL ASSOCIATED WITH FILE SPEC, FORCE TO LOOK LIKE DEVICE NAME AND
                                  0041   152 ; ONLY SUCCEED IF INPUT REPRESENTS DEVICE NAME.
                                  0041   153 ;
                                  0041   154          .ENTRY  -
                         007C     0041   155 LIB$ASSIGN_DEV, ^M<R2,R3,R4,R5,R6>
          5E   000000B0 8F   C2   0043   156          SUBL    #FAB$C_BLN+NAM$C_BLN,SP         ;ALLOCATE ROOM FOR FAB AND NAM
                    56   5E   D0   004A   157          MOVL    SP,R6                           ;GET ADDRESS OF FAB
                         005B   30   004D   158          BSBW    ASSIGN_SETUP                    ;PERFORM COMMON SETUP
              61    50   3A   3A   0050   159 10$:      LOCC    #^A/:/,R0,(R1)                  ;DOES STRING CONTAIN A COLON?
                         11   13   0054   160          BEQL    20$                             ;BRANCH IF NO
                         50   D7   0056   161          DECL    R0                              ;WAS THIS LAST CHARACTER?
                         2A   13   0058   162          BEQL    30$                             ;BRANCH IF YES
                01 A1    3A   91   005A   163          CMPB    #^A/:/,1(R1)                    ;IS THIS A NODE SPECIFIER?
                         24   12   005E   164          BNEQ    30$                             ;BRANCH IF NOT, EXPLICIT ":"
                         51   02   C0   0060   165          ADDL    #2,R1                           ;BUMP PAST "::"
                         50   D7   0063   166          DECL    R0                              ;ONE LESS CHARACTER IN COUNT
                         E9   11   0065   167          BRB     10$                             ;TRY FOR REAL ":"
                         50   62   9A   0067   168 20$:      MOVZBL  (R2),R0                         ;GET SIZE OF STRING
                         50   04   C0   006A   169          ADDL    #4,R0                           ;ROUND UP FOR STACK ALIGNMENT
                         50   03   CA   006D   170          BICL    #3,R0                           ;SIZE NEEDED ON STACK
                         5E   50   C2   0070   171          SUBL    R0,SP                           ;ALLOCATE SPACE FOR COPY
          6E    50   3A   04 B2   62   2C   0073   172          MOVC5   (R2),@4(R2),#^A/:/,R0,(SP)      ;MAKE A COPY WITH A ":" AT END
                34 A6    04 BC   01   81   007A   173          ADDB3   #1,@4(AP),FAB$B_FNS(R6)         ;SET SIZE WITH COLON
                         2C A6   5E   D0   0080   174          MOVL    SP,FAB$L_FNA(R6)                ;SET ADDRESS OF COPY
                      00 04 A6   10   E2   0084   175 30$:      BBSS    #FAB$V_NFS,FAB$L_FOP(R6),35$    ;SET NON FILE STRUCTURE ACCESS
                                  0089   176 35$:      $PARSE  FAB=<(R6)>                      ;GET THE DEVICE INFORMATION
                            15 50   E9   0092   177 40$:      BLBC    R0,50$                  ;BRANCH IF ERROR TO RETURN CODE
                         50   0000'8F 3C   0095   178          MOVZWL  #SS$_IVDEVNAM,R0                ;ASSUME FAILURE
                                       D3   009A   179          BITL    #<NAM$M_EXP_DIR!NAM$M_EXP_NAME!-;CHECK FOR ANYTHING BUT DEVICE
                                  009B   180                  NAM$M_EXP_TYPE!NAM$M_EXP_VER!-
                                  009B   181                  NAM$M_GRP_MBR!NAM$M_WILDCARD>,-
                                  009B   182                  NAM$L_FNB+FAB$C_BLN(R6)
                0084 C6    00080147 8F        009B   183          BNEQ    50$                     ;BRANCH IF ANY OTHERS SET
                         05   12   00A3   184          CALLG   (AP),W^ASSIGN_CHAN              ;CALL FOR STACK CLEANUP
                00F1'CF    6C   FA   00A5   185 50$:      RET
                             04   00AA   186
                                  00AB   187 ;
                                  00AB   188 ;       Routine used for common setup between ASSIGN_FILE and ASSIGN_DEV
                                  00AB   189 ;
```

```
                                00AB   190 ASSIGN_SETUP:
          50   00000000'8F      D0 00AB   191         MOVL    #LIB$_INVARG,R0                 ;ASSUME FAILURE
                       6C   02  91 00B2   192         CMPB    #2,(AP)                         ;ARE THERE 2 ARGUMENTS?
                            39  12 00B5   193         BNEQ    10$                             ;BRANCH IF NOT
                 04 BC   00FF  8F B1 00B7   194         CMPW    #255,@4(AP)                     ;IS SIZE OF DEVNAME REASONABLE?
                            31  19 00BD   195         BLSS    10$                             ;BRANCH IF SIZE TOO BIG
    66   00B0 8F    00    6E   00 2C 00BF   196         MOVC5   #0,(SP),#0,#FAB$C_BLN+NAM$C_BLN,(R6) ;ZERO THEM
                       66   03  90 00C7   197         MOVB    #FAB$C_BID,FAB$B_BID(R6)        ;SET BLOCK ID TO FAB
                01 A6    50 8F  90 00CA   198         MOVB    #FAB$C_BLN,FAB$B_BLN(R6)        ;SET LENGTH OF FAB
                   50 A6    02  90 00CF   199         MOVB    #NAM$C_BID,NAM$B_BID+FAB$C_BLN(R6) ;SET BLOCK ID TO NAME BLOCK
                51 A6    60 8F  90 00D3   200         MOVB    #NAM$C_BLN,NAM$B_BLN+FAB$C_BLN(R6) ;SET LENGTH OF NAME BLOCK
                28 A6    50 A6  9E 00D8   201         MOVAB   FAB$C_BLN(R6),FAB$L_NAM(R6)     ;INDICATE NAME BLOCK EXISTS
                   52    04 AC  D0 00DD   202         MOVL    4(AP),R2                        ;GET ADDRESS OF DESCRIPTOR
                         50   62  9A 00E1   203         MOVZBL  (R2),R0                         ;GET SIZE OF STRING
                      51    04 A2  D0 00E4   204         MOVL    4(R2),R1                        ;GET ADDRESS OF STRING
                   34 A6    50  90 00E8   205         MOVB    R0,FAB$B_FNS(R6)                ;SET SIZE OF DEVICE NAME
                   2C A6    51  D0 00EC   206         MOVL    R1,FAB$L_FNA(R6)                ;SET ADDRESS OF DEVICE NAME
                            05  05 00F0   207 10$:     RSB                                     ;RETURN
                                00F1   208
                                00F1   209 ;
                                00F1   210 ;       Routine used for common closure for both ASSIGN_FILE and ASSIGN_DEV
                                00F1   211 ;
                                00F1   212 ASSIGN_CHAN:
                          0000  00F1   213         .WORD   0                               ;USE COMMON REGISTER SET
                   54    65 A6  9E 00F3   214         MOVAB   NAM$T_DVI+1+FAB$C_BLN(R6),R4    ;SET UP ADDRESS OF DEVICE NAME FIELD
                            54  DD 00F7   215         PUSHL   R4                              ;FORM A DESCRIPTOR
                7E    FF A4  9A 00F9   216         MOVZBL  -1(R4),-(SP)                    ;GET SIZE OF FIELD
                      55   5E  D0 00FD   217         MOVL    SP,R5                           ;R5 IS INPUT DESC FOR $TRNLOG
                64    5F 8F  91 0100   218         CMPB    #^A\_\,(R4)                     ;DO WE ALREADY HAVE A PHYSICAL NAME?
                            30  13 0104   219         BEQL    10$                             ;BRANCH IF YES TO ASSIGN THE CHANNEL
                      5E   32  C2 0106   220         SUBL    #50,SP                          ;ALLOCATE A BUFFER
                           5E  DD 0109   221         PUSHL   SP                              ;SET UP THE ADDRESS
                           32  DD 010B   222         PUSHL   #50                             ;AND SIZE
                      53   5E  D0 010D   223         MOVL    SP,R3                           ;R3 IS OUTPUT DESC FOR $TRNLOG
                               0110   224         $TRNLOG_S (R5),(R3),(R3)                 ;TRANSLATE THE STRING
                        29 50  E9 0123   225         BLBC    R0,20$                          ;BRANCH IF FAILURE
                      55   53  D0 0126   226         MOVL    R3,R5                           ;SET UP TO USE TRANSLATED STRING
                04 B5    1B  91 0129   227         CMPB    #27,@4(R5)                      ;DOES OUTPUT START WITH ESCAPE?
                           07  12 012D   228         BNEQ    10$                             ;BARNCH IF NOT
                      65   04  C2 012F   229         SUBL    #4,(R5)                         ;REDUCE SIZE BY PROCESS PERM STUFF
                04 A5    04  C0 0132   230         ADDL    #4,4(R5)                        ;POINT PAST IT
                           00  DD 0136   231 10$:     PUSHL   #0                              ;AREA TO GET CHANNEL BACK IN
                      52   5E  D0 0138   232         MOVL    SP,R2                           ;GET ADDRESS OF AREA
                               013B   233         $ASSIGN_S (R5),(R2)                      ;ASSIGN A CHANNEL
                        04 50  E9 0148   234         BLBC    R0,20$                          ;BRANCH IF THIS METHOD FAILS ALSO
                08 BC   62  B0 014B   235         MOVW    (R2),@8(AP)                     ;STORE THE CHANNEL FOR THE USER
                            04  05 014F   236 20$:     RET                                     ;RETURN
                                0150   237
                                0150   238         .END
```

```
$$.TMP1                         = 00000001
$$.TMP2                         = 00000066
$$T1                            = 00000001
ASSIGN_CHAN                       000000F1 R      02
ASSIGN_SETUP                      000000AB R      02
FAB$B_BID                       = 00000000
FAB$B_BLN                       = 00000001
FAB$B_FNS                       = 00000034
FAB$C_BID                       = 00000003
FAB$C_BLN                       = 00000050
FAB$L_FNA                       = 0000002C
FAB$L_FOP                       = 00000004
FAB$L_NAM                       = 00000028
FAB$V_NFS                       = 00000010
LIB$ASSIGN                        00000000 RG     02
LIB$ASSIGN_DEV                    00000041 RG     02
LIB$ASSIGN_FILE                   00000020 RG     02
LIB$_INVARG                       ******** X      02
NAM$B_BID                       = 00000000
NAM$B_BLN                       = 00000001
NAM$C_BID                       = 00000002
NAM$C_BLN                       = 00000060
NAM$L_FNB                       = 00000034
NAM$M_EXP_DIR                   = 00000040
NAM$M_EXP_NAME                  = 00000004
NAM$M_EXP_TYPE                  = 00000002
NAM$M_EXP_VER                   = 00000001
NAM$M_GRP_MBR                   = 00080000
NAM$M_WILDCARD                  = 00000100
NAM$T_DVI                       = 00000014
RMS$_CHN                          ******** X    02
RMS$_PRV                          ******** X    02
SS$_IVDEVNAM                      ******** X    02
SYS$ASSIGN                        ******** GX   02
SYS$PARSE                         ******** GX   02
SYS$TRNLOG                        ******** GX   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	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
_LIB$CODE	00000150 (336.)	02 (2.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.09	00:00:00.78
Command processing	107	00:00:00.47	00:00:01.61
Pass 1	171	00:00:03.96	00:00:08.75
Symbol table sort	0	00:00:00.36	00:00:00.61

Pass 2	57	00:00:00.90	00:00:01.86
Symbol table output	6	00:00:00.04	00:00:00.04
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	374	00:00:05.85	00:00:13.69

The working set limit was 1050 pages.
21183 bytes (42 pages) of virtual memory were used to buffer the intermediate code.
There were 20 pages of symbol table space allocated to hold 351 non-local and 12 local symbols.
238 source lines were read in Pass 1, producing 19 object records in Pass 2.
16 pages of virtual memory were used to define 14 macros.

+----------------------------+
! Macro library statistics !
+----------------------------+

Macro library name	Macros defined
_255DUA28:[SYSLIB]STARLET.MLB;2	11

486 GETS were required to define 11 macros.

There were no errors, warnings or information messages.

```
MACRO/DISA=TRACE/LIS=LIS$:LIBASSIGN/OBJ=OBJ$:LIBASSIGN MSRC$:LIBASSIGN/UPDATE=(ENH$:LIBASSIGN)
```

LIBACP
LIS

CALLIMAGE
LIS

DISMOUMSG
LIS

LIBCLICAL
LIS

LIBCQMAT
LIS

FILEDFLT
LIS

LIBASSIGN
LIS

CORMSG
LIS

CVTATB
LIS

LIBCQUAL
LIS

INITMSG
LIS

CREMAPSEC
LIS