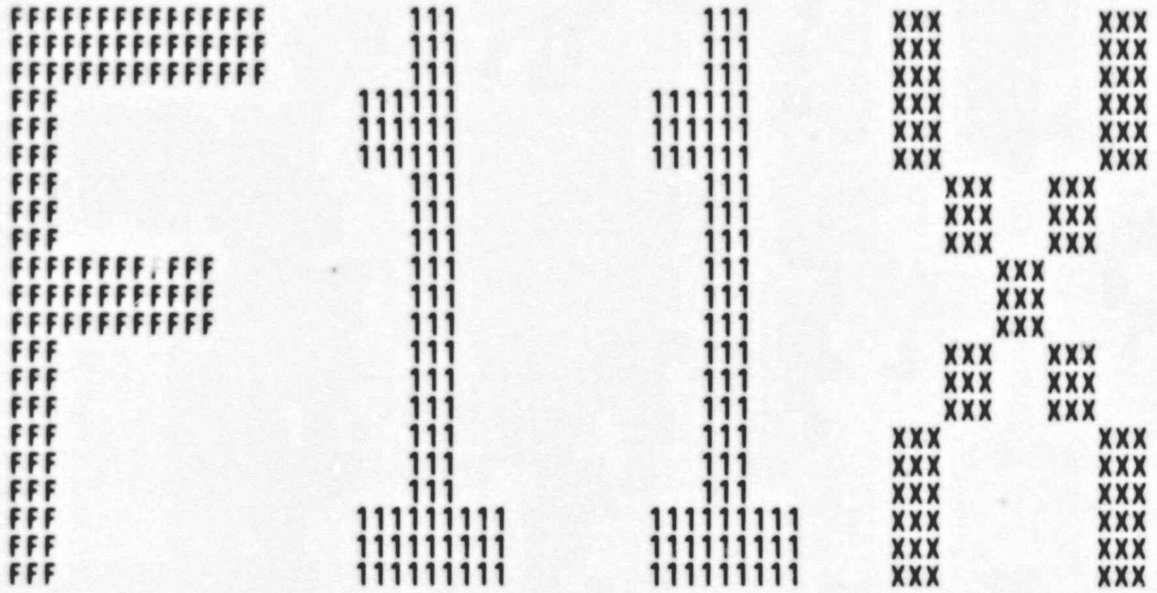

_$2

Symb

IOCS
IO_C
IO_C
IO_D
IO_F
IO_S
KICL

KILL
KILL
LB_B
LB_D
LB_F
LB_K
LB_L
LOCA
LOCA
LOCK

LOCK
LOCK
LOCK
LOC_
LOC_
L_CO
L_CO
L_DA
L_DA
MAIN
MAKE
MAKE
MAKE
MAKE
MAKE
MAKE

MAKE
MAKE
MAP_
MAP_

MAP
MARK
MARK
MARK
MARK
MARK

```
MM      MM    AAAAAA    TTTTTTTTTT    CCCCCCCC  HH      HH  NN      NN    AAAAAA    MM      MM  EEEEEEEEEE
MM      MM    AAAAAA    TTTTTTTTTT    CCCCCCCC  HH      HH  NN      NN    AAAAAA    MM      MM  EEEEEEEEEE
MMMM  MMMM  AA      AA      TT      CC          HH      HH  NN      NN  AA      AA  MMMM  MMMM  EE
MMMM  MMMM  AA      AA      TT      CC          HH      HH  NN      NN  AA      AA  MMMM  MMMM  EE
MM  MM  MM  AA      AA      TT      CC          HH      HH  NNNN    NN  AA      AA  MM  MM  MM  EE
MM  MM  MM  AA      AA      TT      CC          HH      HH  NNNN    NN  AA      AA  MM  MM  MM  EE
MM      MM  AA      AA      TT      CC          HHHHHHHHHH  NN  NN  NN  AA      AA  MM      MM  EEEEEEEE
MM      MM  AA      AA      TT      CC          HHHHHHHHHH  NN  NN  NN  AA      AA  MM      MM  EEEEEEEE
MM      MM  AAAAAAAAAA      TT      CC          HH      HH  NN    NNNN  AAAAAAAAAA  MM      MM  EE
MM      MM  AAAAAAAAAA      TT      CC          HH      HH  NN    NNNN  AAAAAAAAAA  MM      MM  EE
MM      MM  AA      AA      TT      CC          HH      HH  NN      NN  AA      AA  MM      MM  EE
MM      MM  AA      AA      TT      CC          HH      HH  NN      NN  AA      AA  MM      MM  EE
MM      MM  AA      AA      TT        CCCCCCCC  HH      HH  NN      NN  AA      AA  MM      MM  EEEEEEEEEE
MM      MM  AA      AA      TT        CCCCCCCC  HH      HH  NN      NN  AA      AA  MM      MM  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  MATCHNAME       Match General Wild Card Specification
0000     2          .IDENT  'V04-000'
0000     3
0000     4  ;
0000     5  ;*****************************************************************************
0000     6  ;*                                                                           *
0000     7  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     8  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     9  ;*  ALL RIGHTS RESERVED.                                                     *
0000    10  ;*                                                                           *
0000    11  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
0000    12  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
0000    13  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
0000    14  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
0000    15  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
0000    16  ;*  TRANSFERRED.                                                              *
0000    17  ;*                                                                           *
0000    18  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
0000    19  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0000    20  ;*  CORPORATION.                                                              *
0000    21  ;*                                                                           *
0000    22  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
0000    23  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
0000    24  ;*                                                                           *
0000    25  ;*                                                                           *
0000    26  ;*****************************************************************************
0000    27
0000    28  ;++
0000    29  ;
0000    30  ; FACILITY:  Files-11 Structure Level 2
0000    31  ;
0000    32  ; ABSTRACT:
0000    33  ;
0000    34  ;       This routine performs the general embedded wild card matching
0000    35  ;       algorithm.
0000    36  ;
0000    37  ; ENVIRONMENT:
0000    38  ;
0000    39  ;       VAX/VMS Operating System
0000    40  ;
0000    41  ;--
0000    42  ;
0000    43  ;
0000    44  ; AUTHOR:  Andrew C. Goldstein,  CREATION DATE:  10-Aug-1979  11:36
0000    45  ;
0000    46  ; MODIFIED BY:
0000    47  ;
0000    48  ;       V02-001 MLJ0031         Martin L. Jack, 4-Aug-1981  6:32
0000    49  ;                Reorganize for simplicity and speed.
0000    50  ;
0000    51  ;**
```

```
                     0000     53          .SBTTL  FMG$MATCH_NAME, general wild card matching
                     0000     54
                     0000     55 ;++
                     0000     56 ;
                     0000     57 ; Functional Description:
                     0000     58 ;       This routine performs the general embedded wild card matching
                     0000     59 ;       algorithm.
                     0000     60 ;
                     0000     61 ; Calling Sequence:
                     0000     62 ;       JSB
                     0000     63 ;
                     0000     64 ; Input Parameters:
                     0000     65 ;       R2 = Length of candidate string.
                     0000     66 ;       R3 = Address of candidate string.
                     0000     67 ;       R4 = Length of pattern string.
                     0000     68 ;       R5 = Address of pattern string.
                     0000     69 ;
                     0000     70 ; Implicit Inputs:
                     0000     71 ;       none
                     0000     72 ;
                     0000     73 ; Output Parameters:
                     0000     74 ;       none
                     0000     75 ;
                     0000     76 ; Implicit Outputs:
                     0000     77 ;       none
                     0000     78 ;
                     0000     79 ; Routines Called:
                     0000     80 ;       none
                     0000     81 ;
                     0000     82 ; Routine Value:
                     0000     83 ;       True if the strings match.
                     0000     84 ;
                     0000     85 ; Signals:
                     0000     86 ;       none
                     0000     87 ;
                     0000     88 ; Side Effects:
                     0000     89 ;       R1-R5 destroyed.
                     0000     90 ;
                     0000     91 ;--
                     0000     92
                00000000     93          .PSECT  $CODE$,NOWRT,EXE,WORD
                     0000     94
                     0000     95 FMG$MATCH_NAME::
03C0 8F     BB       0000     96          PUSHR   #^M<R6,R7,R8,R9>        ; Save registers
     50     D4       0004     97          CLRL    R0                      ; Assume failure
     56     D4       0006     98          CLRL    R6                      ; Clear saved candidate count
                     0008     99 ;
                     0008    100 ; Main scanning loop.
                     0008    101 ;
     54     D7       0008    102 10$:     DECL    R4                      ; Pattern exhausted?
     24     19       000A    103          BLSS    30$                     ; Branch if yes
51   85     9A       000C    104          MOVZBL  (R5)+,R1                ; Get next character in pattern
2A   51     91       000F    105          CMPB    R1,#^A'*'               ; Pattern specifies wild string?
     28     13       0012    106          BEQL    60$                     ; Branch if yes
     52     D7       0014    107          DECL    R2                      ; Candidate exhausted?
     1F     19       0016    108          BLSS    50$                     ; Branch if yes
83   51     91       0018    109          CMPB    R1,(R3)+                ; Compare pattern to candidate
```

```
                EB   13  001B  110          BEQL    10$                     ; Branch if pattern equals candidate
            25  51   91  001D  111          CMPB    R1,#^A'%'               ; Pattern specifies wild character?
                E6   13  0020  112          BEQL    10$                     ; Branch if yes
                         0022  113 ;
                         0022  114 ; We have detected a mismatch, or we are out of pattern while there is
                         0022  115 ; candidate left.  Back up to the last '*', advance a candidate character,
                         0022  116 ; and try again.
                         0022  117 ;
                56   D7  0022  118 20$:    DECL    R6                      ; Count a saved candidate character
                11   19  0024  119          BLSS    50$                     ; Branch if no saved candidate
                57   D6  0026  120          INCL    R7                      ; Set to try next character
            52  56   7D  0028  121          MOVQ    R6,R2                   ; Restore descriptors to backup point
            54  58   7D  002B  122          MOVQ    R8,R4                   ;
                D8   11  002E  123          BRB     10$                     ; Continue testing
                         0030  124 ;
                         0030  125 ; Here when pattern is exhausted.
                         0030  126 ;
                52   D5  0030  127 30$:    TSTL    R2                      ; Candidate exhausted?
                EE   12  0032  128          BNEQ    20$                     ; Branch if no
                         0034  129 ;
                         0034  130 ; Here to return.
                         0034  131 ;
            50  01   D0  0034  132 40$:    MOVL    #1,R0                   ; Set success return
          03C0  8F   BA  0037  133 50$:    POPR    #^M<R6,R7,R8,R9>        ; Restore registers
                05   003B  134          RSB                             ; Return
                         003C  135 ;
                         003C  136 ; We have detected a '*' in the pattern.  Save the pointers for backtracking.
                         003C  137 ;
                54   D5  003C  138 60$:    TSTL    R4                      ; Pattern null after '*'?
                F4   13  003E  139          BEQL    40$                     ; Branch if yes
            56  52   7D  0040  140          MOVQ    R2,R6                   ; Save descriptors of both strings
            58  54   7D  0043  141          MOVQ    R4,R8                   ;
                C0   11  0046  142          BRB     10$                     ; Continue testing
                         0048  143
                         0048  144          .END
```

```
ACL_TYPE        = 00000007
AQB_TYPE        = 00000005
BITMAP_TYPE     = 00000001
CACHE_TYPE      = 00000006
CHIP_TYPE       = 00000008
DATA_TYPE       = 00000004
DIRECTORY_TYPE  = 00000002
FCB_TYPE        = 00000000
FMG$MATCH_NAME    00000000 RG    01
HEADER_TYPE     = 00000000
INDEX_TYPE      = 00000003
MVL_TYPE        = 00000004
QUOTA_TYPE      = 00000005
RVT_TYPE        = 00000003
VCB_TYPE        = 00000002
WCB_TYPE        = 00000001
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
$CODE$	00000048 (72.)	01 (1.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC WORD

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	30	00:00:00.08	00:00:00.33
Command processing	127	00:00:00.69	00:00:02.53
Pass 1	85	00:00:00.67	00:00:02.66
Symbol table sort	0	00:00:00.00	00:00:00.00
Pass 2	42	00:00:00.44	00:00:01.15
Symbol table output	2	00:00:00.02	00:00:00.02
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	290	00:00:01.93	00:00:06.72

The working set limit was 750 pages.
2825 bytes (6 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 16 non-local and 6 local symbols.
245 source lines were read in Pass 1, producing 11 object records in Pass 2.
2 pages of virtual memory were used to define 2 macros.

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

Macro library name                    Macros defined
------------------                    --------------
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                 0
_$255$DUA28:[SYSLIB]STARLET.MLB;2              0
TOTALS (all libraries)                         0

0 GETS were required to define 0 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:MATCHNAME/OBJ=OBJ$:MATCHNAME MSRC$:FCPPRE/UPDATE=(ENH$:FCPPRE)+MSRC$:MATCHNAME/UPDATE=(ENH$:MATCHNAME)+EXECML$/LIB
```

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

PMS
LIS

QUOTAUTIL
LIS

MAKPTR
LIS

MATCHNAME
LIS

MPWIND
LIS

LOCKDB
LIS

LOCKERS
LIS

MAKACC
LIS

PARSNM
LIS

MAPVBN
LIS

IODONE
LIS

LOCKDN
LIS

MODIFY
LIS

MOUNT
LIS

NXTHDR
LIS

MAKNMB
LIS

MAKSTR
LIS