```
RRRRRRRRRRR      MMM          MMM      SSSSSSSSSSS
RRRRRRRRRRRR     MMM          MMM      SSSSSSSSSSSS
RRRRRRRRRRRR     MMM          MMM      SSSSSSSSSSSS
RRR          RRR MMMMMMM    MMMMMMM   SSS
RRR          RRR MMMMMMM    MMMMMMM   SSS
RRR          RRR MMMMMMM    MMMMMMM   SSS
RRR          RRR MMM   MMM   MMM   MMM SSS
RRR          RRR MMM   MMM   MMM   MMM SSS
RRR          RRR MMM   MMM   MMM   MMM SSS
RRRRRRRRRRR      MMM          MMM      SSSSSSSSS
RRRRRRRRRRRR     MMM          MMM      SSSSSSSSS
RRRRRRRRRRRR     MMM          MMM      SSSSSSSSS
RRR    RRR       MMM          MMM               SSS
RRR    RRR       MMM          MMM               SSS
RRR    RRR       MMM          MMM               SSS
RRR       RRR    MMM          MMM               SSS
RRR       RRR    MMM          MMM               SSS
RRR       RRR    MMM          MMM               SSS
RRR          RRR MMM          MMM  SSSSSSSSSSS
RRR       RRR    MMM          MMM               SSS
RRR          RRR MMM          MMM  SSSSSSSSSSS
RRR          RRR MMM          MMM  SSSSSSSSSSS
RRR          RRR MMM          MMM  SSSSSSSSSSS
```

```
RRRRRRRR    MM       MM    SSSSSSSS    000000    FFFFFFFFFF    SSSSSSSS    CCCCCCCC  NN       NN
RRRRRRRR    MM       MM    SSSSSSSS    000000    FFFFFFFFFF    SSSSSSSS    CCCCCCCC  NN       NN
RR      RR  MMMM   MMMM  SS          00      00  FF          SS          CC          NN       NN
RR      RR  MMMM   MMMM  SS          00      00  FF          SS          CC          NN       NN
RR      RR  MM  MM  MM  SS          00    0000  FF          SS          CC          NNNN     NN
RR      RR  MM  MM  MM  SS          00    0000  FF          SS          CC          NNNN     NN
RRRRRRRR    MM       MM    SSSSSS    00  00  00  FFFFFFFF      SSSSSS    CC          NN  NN   NN
RRRRRRRR    MM       MM    SSSSSS    00  00  00  FFFFFFFF      SSSSSS    CC          NN  NN   NN
RR  RR      MM       MM          SS  0000    00  FF                SS  CC          NN     NNNN
RR  RR      MM       MM          SS  0000    00  FF                SS  CC          NN     NNNN
RR    RR    MM       MM          SS  00      00  FF                SS  CC          NN       NN      ....
RR    RR    MM       MM          SS  00      00  FF                SS  CC          NN       NN      ....
RR      RR  MM       MM  SSSSSSSS    000000    FF          SSSSSSSS    CCCCCCCC  NN       NN      ....
RR      RR  MM       MM  SSSSSSSS    000000    FF          SSSSSSSS    CCCCCCCC  NN       NN      ....

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          $BEGIN  RMSOFSCN,000,RM$RMS,<SCAN FILENAME STRING SYSTEM SERVICE>
0000     2
0000     3  ;*****************************************************************************
0000     4  ;*                                                                           *
0000     5  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     6  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     7  ;*  ALL RIGHTS RESERVED.                                                     *
0000     8  ;*                                                                           *
0000     9  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
0000    10  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
0000    11  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
0000    12  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
0000    13  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
0000    14  ;*  TRANSFERRED.                                                             *
0000    15  ;*                                                                           *
0000    16  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
0000    17  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0000    18  ;*  CORPORATION.                                                             *
0000    19  ;*                                                                           *
0000    20  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
0000    21  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    22  ;*                                                                           *
0000    23  ;*                                                                           *
0000    24  ;*****************************************************************************
0000    25  ;
0000    26  ;++
0000    27  ; FACILITY: RMS32
0000    28  ;
0000    29  ; ABSTRACT:
0000    30  ;                   Routine to perform the $FILESCAN function to
0000    31  ;                   scan a string in order to recognize a file specification.
0000    32  ;
0000    33  ;
0000    34  ; ENVIRONMENT:
0000    35  ;
0000    36  ; AUTHOR: Ron Schaefer,          Creation Date: 13-Apr-1983
0000    37  ;
0000    38  ; Modified By:
0000    39  ;
0000    40  ;       V03-005 JEJ0028         J E Johnson             11-Apr-1984
0000    41  ;               Minor equate file cleanup.
0000    42  ;
0000    43  ;       V03-004 RAS0223         Ron Schaefer            16-Dec-1983
0000    44  ;               Change $SCBDEF and SCB$xxx to $FSCBDEF and FSCB$xxx.
0000    45  ;
0000    46  ;       V03-003 KBT0575         Keith B. Thompson       5-Aug-1983
0000    47  ;               Add new parameter and root directory field
0000    48  ;
0000    49  ;       V03-002 SOP0001         J. R. Sopka             20 July 1983
0000    50  ;               Access LENGTH and POINTER fields of file specification string
0000    51  ;               descriptor separately as WORD and LONGWORD fields respectively.
0000    52  ;               This routine assumes that the descriptor is not a varying length
0000    53  ;               string type which points to a current length count followed by
0000    54  ;               the ASCII text characters.
0000    55  ;
0000    56  ;--
0000    57  ;
```

0000 58

```
0000     60          .SBTTL  DECLARATIONS
0000     61
0000     62 ;
0000     63 ; INCLUDE FILES:
0000     64 ;
0000     65
0000     66 ;
0000     67 ; MACROS:
0000     68 ;
0000     69
0000     70          $DSCDEF         ; define descriptor field offsets and constants
0000     71          $FSCNDEF        ; define item list fields and codes
0000     72          $FSCBDEF        ; define RMS SCAN_STRING scratch block size
0000     73          $SSDEF          ; Define system return codes
0000     74
0000     75 ;
0000     76 ; EQUATED SYMBOLS:
0000     77 ;
0000     78
0000     79 ;
0000     80 ; OWN STORAGE:
0000     81 ;
0000     82
```

```
0000     84          .SBTTL  RMS$FILESCAN - $FILESCAN ROUTINE
0000     85
0000     86 ;++
0000     87 ; RMS$FILESCAN - Scan a string for a file specification
0000     88 ;
0000     89 ;  This module performs the following functions:
0000     90 ;
0000     91 ;       1. Checks for valid input string.
0000     92 ;       2. Calls RM$SCAN_STRING to parse the input string.
0000     93 ;       3. Checks for valid output descriptor list.
0000     94 ;       4. Copies the information returned by RM$SCAN_STRING
0000     95 ;          into the output descriptor list.
0000     96 ;
0000     97 ; CALLING SEQUENCE:
0000     98 ;
0000     99 ;       Entered from EXEC as a result of user's calling SYS$FILESCAN
0000    100 ;
0000    101 ; INPUT PARAMETERS:
0000    102 ;
0000    103 ;       STRDESC   =  Fixed-length type text string descriptor for the
0000    104 ;        4(AP)          file specification string to be parsed.
0000    105 ;
0000    106 ; IMPLICIT INPUTS:
0000    107 ;
0000    108 ;       None
0000    109 ;
0000    110 ; OUTPUT PARAMETERS:
0000    111 ;
0000    112 ;       DESCLST   =  List of item descriptors for the information to be returned.
0000    113 ;        8(AP)          Each descriptor contains an ITEM_CODE field specifying
0000    114 ;                       a portion of the file specification string for which
0000    115 ;                       a LENGTH and ADDR are to be returned.
0000    116 ;
0000    117 ;       FLDFLAGS = Longword to receive flags of which fields are present
0000    118 ;        12(AP)         in the string (optional)
0000    119 ;
0000    120 ;       R1      destroyed
0000    121 ;       R0      status code
0000    122 ;
0000    123 ; IMPLICIT OUTPUTS:
0000    124 ;
0000    125 ;       None
0000    126 ;
0000    127 ; COMPLETION CODES:
0000    128 ;
0000    129 ;       SS$_NORMAL
0000    130 ;       SS$_ACCVIO
0000    131 ;       SS$_BADPARAM
0000    132 ;
0000    133 ; SIDE EFFECTS:
0000    134 ;
0000    135 ;       None
0000    136 ;
0000    137 ;--
0000    138
```

```
                                0000   140          $ENTRY  RMS$FILESCAN
                                0000   141
                     50   04 AC   D0  0000   142          MOVL    4(AP),R0                        ; get input string
                                      0004   143          IFNORD  #DSC$K_S_BLN,(R0),ACCVIO        ; can we read the descriptor?
                          56   60   3C  000A   144          MOVZWL  DSC$W_LENGTH(R0),R6             ; get length field
                     57   04 A0   D0  000D   145          MOVL    DSC$A_POINTER(R0),R7            ; get pointer field
                                      0011   146          IFNORD  R6,(R7),ACCVIO                  ; can we read the string?
                5E   00000104 8F   C2  0017   147          SUBL2   #FSCB$C_BLN,SP                  ; make scratch space
                        5B   5E   D0  001E   148          MOVL    SP,R11                          ; and point at it
                00000000'EF      16  0021   149          JSB     RM$SCAN_STRING                  ; scan the string
                     5A   08 AC   D0  0027   150          MOVL    8(AP),R10                       ; get output desciptor addr
                                2B   13  002B   151          BEQL    DONE                            ; if none, try flags argument
                                      002D   152          IFNOWRT #4,(R10),ACCVIO                 ; can we write the first entry?
                     50   02 AA   3C  0033   153 NXTITM:  MOVZWL  FSCN$W_ITEM_CODE(R10),R0        ; get next item code
                                1F   13  0037   154          BEQL    DONE                            ; zero code means end of list
                                      0039   155          IFNOWRT #FSCN$S_ITEM_LEN,-              ; check access to address field
                                      0039   156                  FSCN$L_ADDR(R10),ACCVIO         ;  and code field of next item
                                      004(   157          CASE    SRC=R0,TYPE=W,LIMIT=#FSCN$_FILESPEC,-
                                      004    158                  DISPLIST=<-
                                      004 )  159                  FILESPEC,-                      ; FSCN$_FILESPEC
                                      00( 0  160                  NODE,-                          ; FSCN$_NODE
                                      00 ,0  161                  DEVICE,-                        ; FSCN$_DEVICE
                                      0( 40  162                  ROOT,-                          ; FSCN$_ROOT
                                      0( 40  163                  DIRECTORY,-                     ; FSCN$_DIRECTORY
                                      0 )40  164                  NAME,-                          ; FSCN$_NAME
                                      0 )40  165                  TYPE,-                          ; FSCN$_TYPE
                                      0 )40  166                  VERSION>                        ; FSCN$_VERSION
                          50   14   D0  0 )54  167          MOVL    #SS$_BADPARAM,R0                ; no such item
                                04   0 )57  168          RET
                                     0 58  169
                     5A   0C AC   D0  0 58  170 DONE:    MOVL    12(AP),R10                      ; get flags longword addr
                                09   13  0( 5   171          BEQL    SUCCESS                         ; branch if not requested
                                      0( 5E  172          IFNOWRT #4,(R10),ACCVIO                 ; can we write it?
                                      0( 64  173
                                      00 4   174          ASSUME  FSCB$V_NODE     EQ      FSCN$V_NODE
                                      00( 4  175          ASSUME  FSCB$V_DEVICE   EQ      FSCN$V_DEVICE
                                      00( 4  176          ASSUME  FSCB$V_ROOT     EQ      FSCN$V_ROOT
                                      006 4  177          ASSUME  FSCB$V_DIRECTORY        EQ      FSCN$V_DIRECTORY
                                      006    178          ASSUME  FSCB$V_NAME     EQ      FSCN$V_NAME
                                      0064   179          ASSUME  FSCB$V_TYPE     EQ      FSCN$V_TYPE
                                      0064   180          ASSUME  FSCB$V_VERSION  EQ      FSCN$V_VERSION
                                      0064   181
                          6A   6B   9A  0064   182          MOVZBL  FSCB$B_FLDFLAGS(R11),(R10)      ; store flags
                                      0067   183 SUCCESS:
                          50   01   D0  0067   184          MOVL    #SS$_NORMAL,R0                  ; success
                                04   006A   185          RET
                                     006B   186
                          50   0C   D0  006B   187 ACCVIO:  MOVL    #SS$_ACCVIO,R0                  ; return accvio
                                04   006E   188          RET
                                     006F   189
                                     006F   190 NULLITEM:
                               6A   B4  006F   191          CLRW    FSCN$W_LENGTH(R10)              ; set no length
                          04   AA   D4  0071   192          CLRL    FSCN$L_ADDR(R10)                ; and no addr
                          5A   08   C0  0074   193 NITEM:   ADDL2   #FSCN$S_ITEM_LEN,R10           ; advance to next descriptor
                             FFB9   31  0077   194          BRW     NXTITM                          ; and continue
                                     007A   195
                                     007A   196 FILESPEC:
```

```
                     6B   95   007A   197          TSTB    FSCB$B_FLDFLAGS(R11)                        ; is any field present?
                     F1   13   007C   198          BEQL    NULLITEM
          6A   04   AB   B0   007E   199          MOVW    FSCB$Q_FILESPEC(R11),FSCN$W_LENGTH(R10) ; set length
    04    AA   08   AB   D0   0082   200          MOVL    FSCB$Q_FILESPEC+4(R11),FSCN$L_ADDR(R10) ; set addr
                     EB   11   0087   201          BRB     NITEM
                               0089   202 NODE:
                     00   E1   0089   203          BBC     #FSCB$V_NODE,-                              ; is this field present?
                E2   6B        008B   204                  FSCB$B_FLDFLAGS(R11),NULLITEM
          6A   0C   AB   B0   008D   205          MOVW    FSCB$Q_NODE(R11),FSCN$W_LENGTH(R10)     ; set length
    04    AA   10   AB   D0   0091   206          MOVL    FSCB$Q_NODE+4(R11),FSCN$L_ADDR(R10)     ; set addr
                     DC   11   0096   207          BRB     NITEM
                               0098   208 DEVICE:
                     01   E1   0098   209          BBC     #FSCB$V_DEVICE,-                                ; is this field pres
                D3   6B        009A   210                  FSCB$B_FLDFLAGS(R11),NULLITEM
          6A   14   AB   B0   009C   211          MOVW    FSCB$Q_DEVICE(R11),FSCN$W_LENGTH(R10)   ; set length
    04    AA   18   AB   D0   00A0   212          MOVL    FSCB$Q_DEVICE+4(R11),FSCN$L_ADDR(R10)   ; set addr
                     CD   11   00A5   213          BRB     NITEM
                               00A7   214
                     02   E1   00A7   215 ROOT:   BBC     #FSCB$V_ROOT,-                              ; is this field present?
                C4   6B        00A9   216                  FSCB$B_FLDFLAGS(R11),NULLITEM
          6A   1C   AB   B0   00AB   217          MOVW    FSCB$Q_ROOT(R11),FSCN$W_LENGTH(R10)     ; set length
    04    AA   20   AB   D0   00AF   218          MOVL    FSCB$Q_ROOT+4(R11),FSCN$L_ADDR(R10)     ; set addr
                     BE   11   00B4   219          BRB     NITEM
                               00B6   220
                               00B6   221 DIRECTORY:
                     03   E1   00B6   222          BBC     #FSCB$V_DIRECTORY,-                         ; is this field present?
                B5   6B        00B8   223                  FSCB$B_FLDFLAGS(R11),NULLITEM
          6A   24   AB   B0   00BA   224          MOVW    FSCB$Q_DIRECTORY(R11),FSCN$W_LENGTH(R10)        ; set length
    04    AA   28   AB   D0   00BE   225          MOVL    FSCB$Q_DIRECTORY+4(R11),FSCN$L_ADDR(R10)        ; set addr
                     AF   11   00C3   226          BRB     NITEM
                               00C5   227
                               00C5   228 NAME:
                     04   E1   00C5   229          BBC     #FSCB$V_NAME,-                              ; is this field present?
                A6   6B        00C7   230                  FSCB$B_FLDFLAGS(R11),NULLITEM
          6A   2C   AB   B0   00C9   231          MOVW    FSCB$Q_NAME(R11),FSCN$W_LENGTH(R10)     ; set length
    04    AA   30   AB   D0   00CD   232          MOVL    FSCB$Q_NAME+4(R11),FSCN$L_ADDR(R10)     ; set addr
                     A0   11   00D2   233          BRB     NITEM
                               00D4   234
                               00D4   235 TYPE:
                     05   E1   00D4   236          BBC     #FSCB$V_TYPE,-                              ; is this field present?
                97   6B        00D6   237                  FSCB$B_FLDFLAGS(R11),NULLITEM
          6A   34   AB   B0   00D8   238          MOVW    FSCB$Q_TYPE(R11),FSCN$W_LENGTH(R10)     ; set length
    04    AA   38   AB   D0   00DC   239          MOVL    FSCB$Q_TYPE+4(R11),FSCN$L_ADDR(R10)     ; set addr
                     91   11   00E1   240          BRB     NITEM
                               00E3   241
                               00E3   242 VERSION:
                     06   E1   00E3   243          BBC     #FSCB$V_VERSION,-                           ; is this field present?
                88   6B        00E5   244                  FSCB$B_FLDFLAGS(R11),NULLITEM
          6A   3C   AB   B0   00E7   245          MOVW    FSCB$Q_VERSION(R11),FSCN$W_LENGTH(R10)  ; set length
    04    AA   40   AB   D0   00EB   246          MOVL    FSCB$Q_VERSION+4(R11),FSCN$L_ADDR(R10)  ; set addr
                     82   11   00F0   247          BRB     NITEM
                               00F2   248
                               00F2   249          .END
```

```
$$.PSECT_EP                     = 00000000
$$RMSTEST                       = 0000001A
$$RMS_PBUGCHK                   = 00000010
$$RMS_TBUGCHK                   = 00000008
$$RMS_UMODE                     = 00000004
ACCVIO                            0000006B R      01
DEVICE                            00000098 R      01
DIRECTORY                         000000B6 R      01
DONE                              00000058 R      01
DSC$A_POINTER                   = 00000004
DSC$K_S_BLN                     = 00000008
DSC$W_LENGTH                    = 00000000
FILESPEC                          0000007A R      01
FSCB$B_FLDFLAGS                 = 00000000
FSCB$C_BLN                      = 00000104
FSCB$Q_DEVICE                   = 00000014
FSCB$Q_DIRECTORY                = 00000024
FSCB$Q_FILESPEC                 = 00000004
FSCB$Q_NAME                     = 0000002C
FSCB$Q_NODE                     = 0000000C
FSCB$Q_ROOT                     = 0000001C
FSCB$Q_TYPE                     = 00000034
FSCB$Q_VERSION                  = 0000003C
FSCB$V_DEVICE                   = 00000001
FSCB$V_DIRECTORY                = 00000003
FSCB$V_NAME                     = 00000004
FSCB$V_NODE                     = 00000000
FSCB$V_ROOT                     = 00000002
FSCB$V_TYPE                     = 00000005
FSCB$V_VERSION                  = 00000006
FSCN$L_ADDR                     = 00000004
FSCN$S_ITEM_LEN                 = 00000008
FSCN$V_DEVICE                   = 00000001
FSCN$V_DIRECTORY                = 00000003
FSCN$V_NAME                     = 00000004
FSCN$V_NODE                     = 00000000
FSCN$V_ROOT                     = 00000002
FSCN$V_TYPE                     = 00000005
FSCN$V_VERSION                  = 00000006
FSCN$W_ITEM_CODE                = 00000002
FSCN$W_LENGTH                   = 00000000
FSCN$_FILESPEC                  = 00000001
NAME                              000000C5 R      01
NITEM                             00000074 R      01
NODE                              00000089 R      01
NULLITEM                          0000006F R      01
NXTITM                            00000033 R      01
RMSSCAN_STRING                    ******** R    X 01
RMS$FILESCAN                    = FFFFFFFE RG     01
ROOT                              000000A7 R      01
SS$_ACCVIO                      = 0000000C
SS$_BADPARAM                    = 00000014
SS$_NORMAL                      = 00000001
SUCCESS                           00000067 R      01
TYPE                              000000D4 R      01
VERSION                           000000E3 R      01
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
RMS$RMS	000000F2 (242.)	01 (1.)	PIC USR CON REL GBL NOSHR EXE RD NOWRT NOVEC BYTE
ABS	00000000 (0.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	30	00:00:00.08	00:00:01.05
Command processing	121	00:00:00.72	00:00:04.08
Pass 1	272	00:00:06.82	00:00:20.12
Symbol table sort	0	00:00:01.09	00:00:01.53
Pass 2	59	00:00:01.26	00:00:04.08
Symbol table output	8	00:00:00.07	00:00:00.08
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	494	00:00:10.07	00:00:30.97

The working set limit was 1200 pages.
36271 bytes (71 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 696 non-local and 2 local symbols.
249 source lines were read in Pass 1, producing 13 object records in Pass 2.
18 pages of virtual memory were used to define 17 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[RMS.OBJ]RMS.MLB;1	4
_255DUA28:[SYS.OBJ]LIB.MLB;1	3
_255DUA28:[SYSLIB]STARLET.MLB;2	6
TOTALS (all libraries)	13

794 GETS were required to define 13 macros.

There were no errors, warnings or information messages.

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

RMS0FSCN
LIS

RM3XALLO
LIS

RMS0CONN
LIS

RMS0ENTER
LIS

RMS0ERASE
LIS

RMS0CREAT
LIS

RMS0DELET
LIS

RMS0FIND
LIS

RM3XSUMO
LIS

RMS0DISPL
LIS

RM3XKEYO
LIS

RMS0GET
LIS

RMS0CLOSE
LIS

RMS0DISC
LIS

RMS0EXTEN
LIS

RMS0BLKIO
LIS