```
RRRRRRRRRRR      UUU          UUU  NNN         NNN     000000000     FFFFFFFFFFFFFFF   FFFFFFFFFFFFFFF
RRRRRRRRRRRR     UUU          UUU  NNN         NNN     000000000     FFFFFFFFFFFFFFF   FFFFFFFFFFFFFFF
RRRRRRRRRRRR     UUU          UUU  NNN         NNN     000000000     FFFFFFFFFFFFFFF   FFFFFFFFFFFFFFF
RRR          RRR UUU          UUU  NNN         NNN  000          000 FFF               FFF
RRR          RRR UUU          UUU  NNN         NNN  000          000 FFF               FFF
RRR          RRR UUU          UUU  NNN         NNN  000          000 FFF               FFF
RRR          RRR UUU          UUU  NNNNNN      NNN  000          000 FFF               FFF
RRR          RRR UUU          UUU  NNNNNN      NNN  000          000 FFF               FFF
RRR          RRR UUU          UUU  NNNNNN      NNN  000          000 FFF               FFF
RRRRRRRRRRR      UUU          UUU  NNN   NNN   NNN  000          000 FFFFFFFFFFF       FFFFFFFFFFF
RRRRRRRRRRRR     UUU          UUU  NNN   NNN   NNN  000          000 FFFFFFFFFFF       FFFFFFFFFFF
RRRRRRRRRRRR     UUU          UUU  NNN   NNN   NNN  000          000 FFFFFFFFFFF       FFFFFFFFFFF
RRR   RRR        UUU          UUU  NNN       NNNNNN 000          000 FFF               FFF
RRR   RRR        UUU          UUU  NNN       NNNNNN 000          000 FFF               FFF
RRR   RRR        UUU          UUU  NNN       NNNNNN 000          000 FFF               FFF
RRR      RRR     UUU          UUU  NNN          NNN 000          000 FFF               FFF
RRR      RRR     UUU          UUU  NNN          NNN 000          000 FFF               FFF
RRR      RRR     UUU          UUU  NNN          NNN 000          000 FFF               FFF
RRR         RRR  UUUUUUUUUUUUUUU  NNN          NNN    000000000     FFF               FFF
RRR         RRR  UUUUUUUUUUUUUUU  NNN          NNN    000000000     FFF               FFF
RRR         RRR  UUUUUUUUUUUUUUU  NNN          NNN    000000000     FFF               FFF
RRR         RRR  UUUUUUUUUUUUUUU  NNN          NNN    000000000     FFF               FFF
RRR         RRR  UUUUUUUUUUUUUUU  NNN          NNN    000000000     FFF               FFF
```

```
_$2

Sym
---
NDX
NDX
NUM
NUM
OPE
OUT
PAC
PAC
PAC
PAC
PAC
PAC
PAC
PAD
PAG
PAG
PAG
PAG
PAG
PAG
PAG
PER
PUT
RCO
RIN

RLI
RNO
RNO
RTY
SAV
STR
STR
STR
STR
STR
STR

STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
```

FILEID**GETDD

```
  GGGGGGGG  EEEEEEEEEE  TTTTTTTTTT  DDDDDDDD    DDDDDDDD
  GGGGGGGG  EEEEEEEEEE  TTTTTTTTTT  DDDDDDDD    DDDDDDDD
GG          EE              TT      DD      DD  DD      DD
GG          EE              TT      DD      DD  DD      DD
GG          EE              TT      DD      DD  DD      DD
GG          EE              TT      DD      DD  DD      DD
GG          EEEEEEEE        TT      DD      DD  DD      DD
GG          EEEEEEEE        TT      DD      DD  DD      DD
GG  GGGGGG  EE              TT      DD      DD  DD      DD
GG  GGGGGG  EE              TT      DD      DD  DD      DD
GG      GG  EE              TT      DD      DD  DD      DD      ....
GG      GG  EE              TT      DD      DD  DD      DD      ....
  GGGGGG    EEEEEEEEEE      TT      DDDDDDDD    DDDDDDDD        ....
  GGGGGG    EEEEEEEEEE      TT      DDDDDDDD    DDDDDDDD        ....


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
```

```
;     1          0001  0 MODULE getdd (  IDENT = 'V04-000'
;     2        P 0002  0                 %BLISS32[, ADDRESSING_MODE (EXTERNAL     = LONG_RELATIVE,
;     3          0003  0                                            NONEXTERNAL  = LONG_RELATIVE)]
;     4          0004  0                 ) =
;     5          0005  1 BEGIN
;     6          0006  1 !
;     7          0007  1 !*****************************************************************************
;     8          0008  1 !*                                                                           *
;     9          0009  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;    10          0010  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;    11          0011  1 !*  ALL RIGHTS RESERVED.                                                      *
;    12          0012  1 !*                                                                           *
;    13          0013  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
;    14          0014  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
;    15          0015  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
;    16          0016  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
;    17          0017  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
;    18          0018  1 !*  TRANSFERRED.                                                              *
;    19          0019  1 !*                                                                           *
;    20          0020  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
;    21          0021  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT     *
;    22          0022  1 !*  CORPORATION.                                                              *
;    23          0023  1 !*                                                                           *
;    24          0024  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
;    25          0025  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
;    26          0026  1 !*                                                                           *
;    27          0027  1 !*                                                                           *
;    28          0028  1 !*****************************************************************************
;    29          0029  1 !
;    30          0030  1 !++
;    31          0031  1 ! FACILITY:     DSR (Digital Standard RUNOFF) / DSRPLUS
;    32          0032  1 !
;    33          0033  1 ! ABSTRACT:     Attempts to read and recognize a display description.
;    34          0034  1 !
;    35          0035  1 ! ENVIRONMENT:  Transportable
;    36          0036  1 !
;    37          0037  1 ! AUTHOR:       R.W.Friday       CREATION DATE: May, 1979
;    38          0038  1 !
;    39          0039  1
```

```
;    41          0040  1 %SBTTL 'Revision History'
;    42          0041  1 !
;    43          0042  1 ! MODIFIED BY:
;    44          0043  1 !
;    45          0044  1 !       003     KFA00003        Ken Alden       15-Sep-1983
;    46          0045  1 !               Changed the error message to RNFIDD "Illegal display descriptor"
;    47          0046  1 !
;    48          0047  1 !       002     RER00002        Ron Randall     07-Mar-1983
;    49          0048  1 !               Global edit of all modules. Updated module names, idents,
;    50          0049  1 !               copyright dates. Changed require files to BLISS library.
;    51          0050  1 !
;    52          0051  1 !--
;    53          0052  1
```

```
;    55          0053  1 %SBTTL 'Module Level Declarations'
;    56          0054  1
;    57          0055  1 !
;    58          0056  1 ! TABLE OF CONTENTS:
;    59          0057  1 !
;    60          0058  1 ! INCLUDE FILES:
;    61          0059  1 !
;    62          0060  1 LIBRARY 'NXPORT:XPORT';                 ! XPORT Library
;    63          0061  1 REQUIRE 'REQ:RNODEF';                   ! RUNOFF variant definitions
;    64          0192  1
;    65        U 0193  1 %IF DSRPLUS %THEN
;    66        U 0194  1 LIBRARY 'REQ:DPLLIB';                   ! DSRPLUS BLISS Library
;    67          0195  1 %ELSE
;    68          0196  1 LIBRARY 'REQ:DSRLIB';                   ! DSR BLISS Library
;    69          0197  1 %FI
;    70          0198  1
;    71          0199  1 !
;    72          0200  1 ! EXTERNAL REFERENCES:
;    73          0201  1 !
;    74          0202  1 EXTERNAL LITERAL                              !Error messages
;    75          0203  1     RNFIDD;
;    76          0204  1
;    77          0205  1 EXTERNAL
;    78          0206  1     FS01 : FIXED_STRING,
;    79          0207  1     IRA : FIXED_STRING;
;    80          0208  1
;    81          0209  1 EXTERNAL ROUTINE
;    82          0210  1     ERMS,
;    83          0211  1     GSLU,
;    84          0212  1     RSKIPS;
;    85          0213  1
```

```
;    87          0214  1 GLOBAL ROUTINE GETDD (DD_VALUE) =
;    88          0215  1
;    89          0216  1 !++
;    90          0217  1 ! FUNCTIONAL DESCRIPTION:
;    91          0218  1 !
;    92          0219  1 !       See the ABSTRACT for a general description.
;    93          0220  1 !
;    94          0221  1 ! FORMAL PARAMETERS:
;    95          0222  1 !
;    96          0223  1 !       DD_VALUE gets set to the value of the recognized display description.
;    97          0224  1 !
;    98          0225  1 ! IMPLICIT INPUTS:
;    99          0226  1 !
;   100          0227  1 !       Implicit input is the one-for-one correspondence
;   101          0228  1 !       between the list of valid display descriptors in the
;   102          0229  1 !       list DD, below, and the literals TCONVRT_XXXX_D
;   103          0230  1 !       defined in the file CONVRT.REQ.
;   104          0231  1 !       This makes it possible to use the index into DD as
;   105          0232  1 !       the display code that gets saved.
;   106          0233  1 !
;   107          0234  1 ! IMPLICIT OUTPUTS:     None
;   108          0235  1 !
;   109          0236  1 ! ROUTINE VALUE:
;   110          0237  1 ! COMPLETION CODES:
;   111          0238  1 !
;   112          0239  1 !       If a valid display description is recognized, a completion code value of +1 is returned.
;   113          0240  1 !       The value zero is returned if no display description is recognized.  The value
;   114          0241  1 !       -1 indicates that the display description could not be recognized; in this case an
;   115          0242  1 !       error message is generated too.
;   116          0243  1 !
;   117          0244  1 ! SIDE EFFECTS:         None
;   118          0245  1 !--
;   119          0246  1
;   120          0247  2     BEGIN
;   121          0248  2     LOCAL
;   122          0249  2         DISPLAY_CODE,
;   123          0250  2         GSLU_RESULT;
;   124          0251  2
;   125          0252  2     !Skip spaces and tabs before the display descriptor.
;   126          0253  2     RSKIPS (IRA);
;   127          0254  2
;   128          0255  2     !Now attempt to get a display descriptor.
;   129          0256  2     !First, initialize FS01, where the results will be returned.
;   130          0257  2     FS_INIT (FS01);
;   131          0258  2     !And now actually try to get the discriptor.
;   132          0259  2     GSLU_RESULT = GSLU (IRA, FS01);
;   133          0260  2
;   134          0261  2     !Distinguish between missing display code and one that
;   135          0262  2     !is given.
;   136          0263  2     IF  .GSLU_RESULT EQL GSLU_NONE
;   137          0264  2     THEN
;   138          0265  2         RETURN 0
;   139          0266  2     ELSE
;   140          0267  2         !Try to recognize the display code supplied.
;   141          0268  3         BEGIN
;   142          0269  3         BIND
;   143          0270  3             DD = UPLIT .CH$PTR(UPLIT('D  ')),           !Decimal
```

```
;  144          0271  3                        CH$PTR(UPLIT('* ')),            !can't be gotten at.
;  145          0272  3                        CH$PTR(UPLIT('LU')),            !All uppercase letters
;  146          0273  3                        CH$PTR(UPLIT('LL')),            !All lowercase letters
;  147          0274  3                        CH$PTR(UPLIT('LM')),            !Mixed case letters.
;  148          0275  3                        CH$PTR(UPLIT('RU')),            !Upper case Roman numerals
;  149          0276  3                        CH$PTR(UPLIT('RL')),            !Lower case Roman numerals
;  150          0277  3                        CH$PTR(UPLIT('RM')),            !Mixed case Roman numerals.
;  151          0278  3                        CH$PTR(UPLIT('O ')),            !Octal
;  152          0279  3                        CH$PTR(UPLIT('H ')) )           !Hex
;  153          0280  3                                            : VECTOR;
;  154          0281  3
;  155          0282  3              !Assume the display name won't be found (i.e., illegal)
;  156          0283  3              DISPLAY_CODE = -1;
;  157          0284  3
;  158          0285  3              !Now, try to recognize the given code.
;  159          0286  3              !If the code is not recognized DISPLAY_CODE will contain the value
;  160          0287  3              !-1.  This error is caught immediately afterwards.
;  161          0288  3              INCR I FROM 0 TO 9 DO
;  162          0289  3
;  163          0290  3                  IF  CH$EQL (.FS_LENGTH(FSO1),
;  164          0291  3                              .FS_START(FSO1),
;  165          0292  3                              2,
;  166          0293  3                              .DD [.I],
;  167          0294  3                              %C' ')
;  168          0295  3                  THEN
;  169          0296  3                      !User's display name has been recognized
;  170          0297  4                      BEGIN
;  171          0298  4                      !Remember the index into the table.  It's the display code.
;  172          0299  4                      DISPLAY_CODE = .I;
;  173          0300  4                      EXITLOOP
;  174          0301  3                      END;
;  175          0302  2              END;
;  176          0303  2
;  177          0304  2              !Make sure a valid display code will be set.
;  178          0305  2              IF  .DISPLAY_CODE EQL -1
;  179          0306  2              THEN
;  180          0307  2                  !User gave an invalid display name
;  181          0308  3                  BEGIN
;  182          0309  3                  ERMS (RNFIDD, .fs_start(FSO1), .fs_length(FSO1));
;  183          0310  3                  RETURN -1
;  184          0311  2                  END;
;  185          0312  2
;  186          0313  2              !Save value of display description, and return to user
;  187          0314  2              .DD_VALUE = .DISPLAY_CODE;
;  188          0315  2              RETURN 1
;  189          0316  2
;  190          0317  1              END;                        !End of GETDD


                                                    .TITLE  GETDD
                                                    .IDENT  \V04-000\

                                                    .PSECT  $PLIT$,NOWRT,NOEXE,2

                                00  00  20  44  00000 P.AAB:  .ASCII  \D \<0><0>                                    ;
                                00  00  20  2A  00004 P.AAC:  .ASCII  \* \<0><0>                                  ;
                                00  00  55  4C  00008 P.AAD:  .ASCII  \LU\<0><0>                                  ;
```

```
                                          00  00  4C  4C  0000C P.AAE:   .ASCII   \LL\<0><0>                              ;
                                          00  00  4D  4C  00010 P.AAF:   .ASCII   \LM\<0><0>                              ;
                                          00  00  55  52  00014 P.AAG:   .ASCII   \RU\<0><0>                              ;
                                          00  00  4C  52  00018 P.AAH:   .ASCII   \RL\<0><0>                              ;
                                          00  00  4D  52  0001C P.AAI:   .ASCII   \RM\<0><0>                              ;
                                          00  00  20  4F  00020 P.AAJ:   .ASCII   \O \<0><0>                              ;
                                          00  00  20  48  00024 P.AAK:   .ASCII   \H \<0><0>                              ;
00000000' 00000000' 00000000' 0000000C' 00000000' 00000000' 00028 P.AAA:   .ADDRESS P.AAB, P.AAC, P.AAD, P.AAE, P.AAF, -     ;
                    00000000' 00000000' 00000000' 00000000' 00040                     P.AAG, P.AAH, P.AAI, P.AAJ, P.AAK       ;

                                                               DD=                 P.AAA
                                                                         .EXTRN   RNFIDD, FS01, IRA
                                                                         .EXTRN   ERMS, GSLU, RSKIPS

                                                                         .PSECT   $CODE$,NOWRT,2

                                                  00FC  00000            .ENTRY   GETDD, Save R2,R3,R4,R5,R6,R7           ; 0214
                                   57 00000000G  EF  9E  00002            MOVAB    IRA, R7                                 ;
                                   56 00000000G  EF  9E  00009            MOVAB    FS01, R6                                ;
                                                 57  DD  00010            PUSHL    R7                                      ; 0253
                          00000000G  EF          01  FB  00012            CALLS    #1, RSKIPS                              ;
                                             0C  A6  D4  00019            CLRL     FS01+12                                 ; 0257
                                   66        10  A6  9E  0001C            MOVAB    FS01+16, FS01                           ;
                               04  A6            66  D0  00020            MOVL     FS01, FS01+4                            ;
                                                 56  DD  00024            PUSHL    R6                                      ; 0259
                                                 57  DD  00026            PUSHL    R7                                      ;
                          00000000G  EF          02  FB  00028            CALLS    #2, GSLU                                ;
                                     02          50  D1  0002F            CMPL     GSLU_RESULT, #2                         ; 0263
                                                 47  13  00032            BEQL     5$                                      ;
                                     55          01  CE  00034            MNEGL    #1, DISPLAY_CODE                        ; 0283
                                                 54  D4  00037            CLRL     I                                       ; 0291
                                     50 00000000'EF44  D0  00039 1$:       MOVL     DD[I], R0                               ; 0293
        02             20          00  B6        0C  A6  2D  00041            CMPC5    FS01+12, @FS01, #32, #2, (R0)           ; 0290
                                                 60      00048                                                             ;
                                                 05  12  00049            BNEQ     2$                                      ;
                                     55          54  D0  0004B            MOVL     I, DISPLAY_CODE                         ; 0299
                                                 04  11  0004E            BRB      3$                                      ; 0297
                          E5         54          09  F3  00050 2$:         AOBLEQ   #9, I, 1$                               ; 0290
                          FFFFFFFF   8F          55  D1  00054 3$:         CMPL     DISPLAY_CODE, #-1                       ; 0305
                                                 16  12  0005B            BNEQ     4$                                      ;
                                             0C  A6  DD  0005D            PUSHL    FS01+12                                 ; 0309
                                                 66  DD  00060            PUSHL    FS01                                    ;
                                      00000000G  8F  DD  00062            PUSHL    #RNFIDD                                 ;
                          00000000G  EF          03  FB  00068            CALLS    #3, ERMS                                ;
                                     50          01  CE  0006F            MNEGL    #1, R0                                  ; 0310
                                                     04  00072            RET                                              ;
                               04  BC            55  D0  00073 4$:         MOVL     DISPLAY_CODE, @DD_VALUE                 ; 0314
                                     50          01  D0  00077            MOVL     #1, R0                                  ; 0315
                                                     04  0007A            RET                                              ;
                                                 50  D4  0007B 5$:         CLRL     R0                                      ; 0317
                                                     04  0007D            RET                                              ;

; Routine Size:  126 bytes,    Routine Base:  $CODE$ + 0000


;   191          0318   1
;   192          0319   1 END                                     !End of module
```

```
;  193          0320  0 ELUDOM
```

```
;                                  PSECT SUMMARY
;
;       Name                     Bytes                         Attributes
;
;  $PLIT$                           80  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                          126  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
```

```
;                           Library Statistics
;
;                                              -------- Symbols --------      Pages       Processing
;       File                                   Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]XPORT.L32;1              590        0         0         252         00:00.1
;  _$255$DUA28:[RUNOFF.SRC]DSRLIB.L32;1        1248        9         0          86         00:00.3
```

```
;                                  COMMAND QUALIFIERS
;
;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:GETDD/OBJ=OBJ$:GETDD MSRC$:GETDD/UPDATE=(ENH$:GETDD)
;
; Size:          126 code + 80 data bytes
; Run Time:         00:04.2
; Elapsed Time:     00:15.6
; Lines/CPU Min:     4528
; Lexemes/CPU-Min: 12778
; Memory Used:  46 pages
; Compilation Complete
```