```
VVV       VVV MMM       MMM SSSSSSSSSSS LLL               IIIIIIIIII BBBBBBBBBBB
VVV       VVV MMM       MMM SSSSSSSSSSS LLL               IIIIIIIIII BBBBBBBBBBB
VVV       VVV MMM       MMM SSSSSSSSSSS LLL               IIIIIIIIII BBBBBBBBBBB
VVV       VVV MMMMM   MMMMM 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 MMMMM 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      MMM       MMM SSSSSSSSSSS LLLLLLLLLLLLLLL IIIIIIIIII BBBBBBBBBBB
     VVV      MMM       MMM SSSSSSSSSSS LLLLLLLLLLLLLLL IIIIIIIIII BBBBBBBBBBB
     VVV      MMM       MMM SSSSSSSSSSS LLLLLLLLLLLLLLL IIIIIIIIII BBBBBBBBBBB
```

```
DDDDDDD   VV      VV   IIIIII  TTTTTTTTTT    AAAAAA   BBBBBBBB  LL          EEEEEEEEEE
DDDDDDD   VV      VV   IIIIII  TTTTTTTTTT    AAAAAA   BBBBBBBB  LL          EEEEEEEEEE
DD    DD   VV    VV      II        TT       AA    AA  BB    BB  LL          EE
DD    DD   VV    VV      II        TT       AA    AA  BB    BB  LL          EE
DD    DD   VV    VV      II        TT       AA    AA  BB    BB  LL          EE
DD    DD   VV    VV      II        TT       AA    AA  BBBBBBB   LL          EEEEEEE
DD    DD   VV    VV      II        TT       AA    AA  BBBBBBB   LL          EEEEEEE
DD    DD   VV    VV      II        TT      AAAAAAAAAA BB    BB  LL          EE
DD    DD   VV    VV      II        TT      AAAAAAAAAA BB    BB  LL          EE
DD    DD    VV  VV       II        TT       AA    AA  BB    BB  LL          EE
DD    DD    VV  VV       II        TT       AA    AA  BB    BB  LL          EE
DDDDDDD      VV        IIIIII      TT       AA    AA  BBBBBBB   LLLLLLLLLL  EEEEEEEEEE  ....
DDDDDDD      VV        IIIIII      TT       AA    AA  BBBBBBB   LLLLLLLLLL  EEEEEEEEEE  ....


MM      MM   AAAAAA   RRRRRRR
MM      MM   AAAAAA   RRRRRRR
MMMM  MMMM  AA    AA  RR    RR
MMMM  MMMM  AA    AA  RR    RR
MM  MM  MM  AA    AA  RR    RR
MM      MM  AA    AA  RRRRRRRR
MM      MM  AA    AA  RRRRRRR
MM      MM AAAAAAAAAA RR  RR
MM      MM AAAAAAAAAA RR  RR
MM      MM  AA    AA  RR   RR
MM      MM  AA    AA  RR   RR
MM      MM  AA    AA  RR    RR
MM      MM  AA    AA  RR    RR
```

```
        .IDENT  'V04-000'

;*******************************************************************************
;*                                                                            *
;*   COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;*   DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
;*   ALL RIGHTS RESERVED.                                                     *
;*                                                                            *
;*   THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
;*   ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
;*   INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
;*   COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
;*   OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
;*   TRANSFERRED.                                                             *
;*                                                                            *
;*   THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
;*   AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
;*   CORPORATION.                                                             *
;*                                                                            *
;*   DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
;*   SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
;*                                                                            *
;*                                                                            *
;*******************************************************************************


; ENVIRONMENT: macro library

; AUTHOR: Ken Henderson        CREATION DATE: 23 Feb 1983

; MODIFIED BY:

;       V03-021 CWH3021         CW Hobbs                 29-Jul-1984
;               Use uppercase letters for spare codes for shadow volumes,
;               F$GETDVI is sensitive to case.

;       V03-020 CWH3020         CW Hobbs                 24-Jul-1984
;               Add several spare codes for shadow volumes.

;       V03-019 LY0507          Larry Yetto              11-JUL-1984 17:34
;               Add MEDIA_ID item code for NONdecoded value from UCB$L_MEDIA_ID

;       V03-018 LY0503          Larry Yetto              10-JUL-1984 10:18
;               Add MEDIA_NAME and MEDIA_TYPE item codes for the decoded
;               strings from UCB$L_MEDIA_ID

;       V03-017 EMD0095         Ellen M. Dusseault       01-May-1984
;               Add DVI$_TT_DECCRT2 item code.

;       V03-016 LMP0221         L. Mark Pilant,          30-Mar-1984  16:49
;               Move the owner UIC and protection from the UCB to the ORB.

;       V03-015 MHB0106         Mark Bramhall            1-Mar-1984
;               Add DVI$_TT_PHYDEVNAM item code.
```

V03-014 CWH3014   CW Hobbs   19-Feb-1984
        Add item codes for dual-path and shadow-set attributes

V03-013 TCM0004   Trudy C. Matthews   04-Jan-1984
        Correct the order of the UCB$L_DEVDEPND2 bit definitions for
        terminal UCBs; the item codes did not correspond to the
        DVI$_xxx values defined for some of the bits (ANSICRT, REGIS,
        BLOCK, AVO, EDIT, DECCRT, and SYSPWD).

V03-012 TCM0003   Trudy C. Matthews   09-Dec-1983
        Change ALLDEVNAM from a hexidecimal string to a padded
        ascii string.

V03-011 KFH0007   Ken Henderson   10 Sep 1983
        Add DEVLOCKNAM and VOLSETMEM.
        ALLDEVNAM: change dtype to HEXSTR
        and outlen to 16.
        Change length of FULLDEVNAM to 16.

V03-010 KFH0006   Ken Henderson   20 Aug 1983
        Prefix bit definitions from TTDEF and TT2DEF
        with "TT_" to indicate device dependence.

V03-009 KFH0005   Ken Henderson   18 Aug 1983
        Remove bunches of bit definitions (STS, DEVSTS)that were
        from SYSDEF.SDL - they are system-private.
        Make LOGVOLNAM applicable to ANY device.
        Remove HOLDSCREEN, DCL_CTRLC, DCL_OUTBND, XON

V03-008 TCM0002   Trudy C. Matthews   24-Jun-1983
        Add GETDVI item code ALLDEVNAM -- allocation class plus
        device name.

V03-007 TCM0001   Trudy C. Matthews   20-Jun-1983
        Add GETDVI item code LOCKID.

V03-006 KFH0004   Ken Henderson   29 Apr 1983
        Add GETDVI item code FULLDEVNAM.
        Also changed MOUNTCNT to ANY class from DISK class.

V03-005 ROW0171   Ralph O. Weber   12-APR-1983
        Add GETDVI item code DEVCHAR2 for the second device
        characteristics longword, UCB$L_DEVCHAR2. Although this
        longword immediately follows UCB$L_DEVCHAR in the UCB, its
        value is returned seperately. This conforms to the precident
        set by UCB$L_DEVDEPEND2 and prevents GETDVI from returning any
        quadword values.

V03-004 KFH0003   Ken Henderson   11 Mar 1983
        Added temporary fix to PAGE item-code.

V03-003 KFH0002   Ken Henderson   9 Mar 1983
        Added STS and DEVSTS and their bitfield
        item-codes.

    V03-002 CWH1002   CW Hobbs   1-Mar-1983
            Make DVI$_PID and DVI$_ACPPID special items.

    V03-001 KFH0001   Ken Henderson   23 Feb 1983
            Added DEVCHAR bitfield item-codes.

```
        .MACRO  DVI_GENERATE_TABLE
;++
; ABSTRACT:
;
; DVI_GENERATE_TABLE macro
;
; This macro expands to generate multiple calls to the DVI_ITEM_CODE macro,
; which must be previously locally defined in the module which invokes
; DVI_GENERATE_TABLE.
;
; The parameters that are passed to the DVI_ITEM_CODE macro follow:
;
;       NAME            is the name of the SYS$GETDVI item-code.
;                       The legal parameter values here are determined by
;                       the $DVIDEF macro (in [VMSLIB.SRC]STARDEFAE.SDL).
;
;       SPECIAL         determines if special handling is required.
;                       The legal parameter values here are: T and F.
;
;       SOURCE          is the offset of the data in the I/O data structure.
;
;       DTYPE           is both a datatype and a usage indicator.
;                       The legal values and examples for this parameter follow:
;
;                       STDTIM          (CTL$GQ_LOGIN)          64 bit time
;                       STDUIC          (PCB$L_UIC)             user ID code
;                       HEXNUM          (CTL$AQ_EXCVEC)         hex number
;                       DECNUM          (PCB$L_BYTLM)           decimal number
;                       PRVMSK          (PHD$Q_PRIVMSK)         privilege mask
;                       PRTMSK          (ORC$W_PROT)            protection mask
;                       STRDSC          (CTL$GL_IMGHDRBF)       string descr
;                       CNTSTR          (PCB$T_TERMINAL)        counted string
;                       PADSTR          (JIB$T_ACCOUNT)         blank padded str
;                       BITVEC          (UCB$L_STS)             bit vector
;                       BITVAL          (UCB$V_ONLINE)          boolean quantity
;                       ACPTYP          (AQB$B_ACPTYP)          ACP type
;
;       BITPOS          is the bit position for bitfield data items.
;
;       OUTLEN          is used by EXE$GETDVI in fetching information (number
;                       of bytes).
;
;       STRUCT          is the user's data structure containing the information.
;
;       DEVTYP          is a flag which indicates disk devices. Legal values
;                       are: ANY, and DISK.
;--
;
```

D 10

```
;*******************************************************************
; The item codes down to MAXBLOCK parallel the DEVTAB in SYSGETDVI.
; DO NOT REORDER OR SEPARATE ANY OF THESE CODES - ADD AT END ONLY!
;*******************************************************************
;

;NAME,  SPECIAL,SOURCE,          DTYPE,  BITPOS,        OUTLEN, STRUCT, DEVTYP
;

; DEVCHAR - Device characteristics
DVI_ITEM_CODE -
DEVCHAR, F,    L_DEVCHAR,        BITVEC, 0,             4,      UCB,    ANY

; DEVCLASS - Device Class
DVI_ITEM_CODE -
DEVCLASS, F,   B_DEVCLASS,       DECNUM, 0,             1,      UCB,    ANY

; DEVTYPE - Device Type
DVI_ITEM_CODE -
DEVTYPE, F,    B_DEVTYPE,        DECNUM, 0,             1,      UCB,    ANY

; DEVBUFSIZ - Width
DVI_ITEM_CODE -
DEVBUFSIZ, F,  W_DEVBUFSIZ,      DECNUM, 0,             2,      UCB,    ANY

;
;NAME,  SPECIAL,SOURCE,          DTYPE,  BITPOS,        OUTLEN, STRUCT, DEVTYP
;

; DEVDEPEND - Device Geometry
DVI_ITEM_CODE -
DEVDEPEND, F,  L_DEVDEPEND,      BITVEC, 0,             4,      UCB,    ANY

; UNIT - Binary unit number
DVI_ITEM_CODE -
UNIT,   F,     W_UNIT,           DECNUM, 0,             2,      UCB,    ANY

; PID - Owner Process ID
DVI_ITEM_CODE -
PID,    T,     PID,              HEXNUM, 0,             4,      UCB,    ANY

;
;NAME,  SPECIAL,SOURCE,          DTYPE,  BITPOS,        OUTLEN, STRUCT, DEVTYP
;

; OWNER - Owner UIC
DVI_ITEM_CODE -
OWNUIC, F,     L_OWNER,          STDUIC, 0,             4,      ORB,    ANY

; PROT - Volume Protection
DVI_ITEM_CODE -
VPROT,  F,     W_PROT,           PRTMSK, 0,             2,      ORB,    ANY

; ERRCNT - Error Count
```

```
DVI_ITEM_CODE -
ERRCNT, F,      W_ERRCNT,       DECNUM, 0,          2,      UCB,    ANY

;NAME,  SPECIAL,SOURCE,         DTYPE,  BITPOS,     OUTLEN, STRUCT, DEVTYP
;

; OPCNT - Operation Count
DVI_ITEM_CODE -
OPCNT,  F,      L_OPCNT,        DECNUM, 0,          4,      UCB,    ANY

; RECSIZ - Blocked record size
DVI_ITEM_CODE -
RECSIZ, F,      W_RECORDSZ,     DECNUM, 0,          2,      VCB,    ANY

; MAXBLOCK - Max Blocks on Vol
DVI_ITEM_CODE -
MAXBLOCK, F,    L_MAXBLOCK,     DECNUM, 0,          4,      UCB,    DISK

;NAME,  SPECIAL,SOURCE,         DTYPE,  BITPOS,     OUTLEN, STRUCT, DEVTYP
;

; DEVDEPEND2 - Terminal bits
DVI_ITEM_CODE -
DEVDEPEND2, F, L_DEVDEPND2,     BITVEC, 0,          4,      UCB,    ANY

; REFCNT - Reference count on UCB
DVI_ITEM_CODE -
REFCNT, F,      W_REFC,         DECNUM, 0,          2,      UCB,    ANY

; DEVNAM - Device Name String
DVI_ITEM_CODE -
DEVNAM, T,      DEVNAM,         PADSTR, 0,          4,      UCB,    ANY

;NAME,  SPECIAL,SOURCE,         DTYPE,  BITPOS,     OUTLEN, STRUCT, DEVTYP
;

; VOLNAM - Volume Name (also known as the LABEL)
DVI_ITEM_CODE -
VOLNAM, T,      VOLNAM,         PADSTR, 0,          4,      VCB,    ANY

; SECTORS - Geometry of disk
DVI_ITEM_CODE -
SECTORS, F,     B_SECTORS,      DECNUM, 0,          1,      UCB,    DISK

; TRACKS - Geometry of disk
DVI_ITEM_CODE -
TRACKS, F,      B_TRACKS,       DECNUM, 0,          1,      UCB,    DISK

;NAME,  SPECIAL,SOURCE,         DTYPE,  BITPOS,     OUTLEN, STRUCT, DEVTYP
;

; CYLINDERS - Geometry of disk
```

```
DVI_ITEM_CODE -
CYLINDERS, F,   W_CYLINDERS,     DECNUM, 0,              2,    UCB,    DISK

; FREEBLOCKS - Count of free blocks on disk
DVI_ITEM_CODE -
FREEBLOCKS, T, FREEBLOCKS,       DECNUM, 0,              4,    VCB,    DISK

; LOGVOLNAM - logical volume name
DVI_ITEM_CODE -
LOGVOLNAM, T,  LOGVOLNAM,        CNTSTR, 0,              4,    VCB,    ANY


;NAME,  SPECIAL,SOURCE,          DTYPE, BITPOS,       OUTLEN, STRUCT, DEVTYP
;

; VOLNUMBER - volume number of this volume
DVI_ITEM_CODE -
VOLNUMBER, T,   VOLNUMBER,       DECNUM, 0,              4,    VCB,    DISK

; VOLCOUNT - Number of volumes in vol set
DVI_ITEM_CODE -
VOLCOUNT, F,    VOLCOUNT,        DECNUM, 0,              0,    RVT,    DISK

; ROOTDEVNAM - Device name of 1st vol in set
DVI_ITEM_CODE -
ROOTDEVNAM, F, ROOTDEVNAM,       PADSTR, 0,              0,    RVT,    DISK


;NAME,  SPECIAL,SOURCE,          DTYPE, BITPOS,       OUTLEN, STRUCT, DEVTYP
;

; NEXTDEVNAM - Next device name in vol set
DVI_ITEM_CODE -
NEXTDEVNAM, F, NEXTDEVNAM,       PADSTR, 0,              0,    RVT,    DISK

; TRANSCNT - Transaction count on volume
DVI_ITEM_CODE -
TRANSCNT, F,    W_TRANS,         DECNUM, 0,              2,    VCB,    ANY

; MOUNTCNT - Mount count for volume
DVI_ITEM_CODE -
MOUNTCNT, F,    W_MCOUNT,        DECNUM, 0,              2,    VCB,    ANY


;NAME,  SPECIAL,SOURCE,          DTYPE, BITPOS,       OUTLEN, STRUCT, DEVTYP
;

; CLUSTER - Allocation cluster
DVI_ITEM_CODE -
CLUSTER, F,     W_CLUSTER,       DECNUM, 0,              2,    VCB,    DISK

; MAXFILES - Maximum files on volume
DVI_ITEM_CODE -
MAXFILES, F,    L_MAXFILES,      DECNUM, 0,              4,    VCB,    DISK

; SERIALNUM - Volume serial number
```

```
DVI_ITEM_CODE -
SERIALNUM, F,   L_SERIALNUM,    DECNUM, 0,              4,      VCB,    DISK

;NAME,  SPECIAL,SOURCE,         DTYPE, BITPOS,          OUTLEN, STRUCT, DEVTYP
;

; ACPPID - ACP Process ID
DVI_ITEM_CODE -
ACPPID, T,      ACPPID,         HEXNUM, 0,              4,      AQB,    ANY

; ACPTYPE - ACP type
DVI_ITEM_CODE -
ACPTYPE, F,     B_ACPTYPE,      ACPTYP, 0,              1,      AQB,    ANY

; Device is a concealed device - BOOLEAN - 1 byte
DVI_ITEM_CODE -
CONCEALED, T,   L_DEVCHAR,      BITVAL, 0,              1,      UCB,    ANY

;
; THE FOLLOWING CODES ARE THE INDIVIDUAL BITS OF THE DEVCHAR LONGWORD
;

;
;NAME,  SPECIAL,SOURCE,         DTYPE, BITPOS,          OUTLEN, STRUCT, DEVTYP
;

; DEVICE RECORD ORIENTED
DVI_ITEM_CODE -
REC,    F,      L_DEVCHAR,      BITVAL, DEV$V_REC,      1,      UCB,    ANY

; CARRIAGE CONTROL DEVICE
DVI_ITEM_CODE -
CCL,    F,      L_DEVCHAR,      BITVAL, DEV$V_CCL,      1,      UCB,    ANY

; DEVICE IS A TERMINAL
DVI_ITEM_CODE -
TRM,    F,      L_DEVCHAR,      BITVAL, DEV$V_TRM,      1,      UCB,    ANY

; DEVICE IS DIRECTORY STRUCTURED
DVI_ITEM_CODE -
DIR,    F,      L_DEVCHAR,      BITVAL, DEV$V_DIR,      1,      UCB,    ANY

; DEVICE IS SINGLE DIRECTORY STRUCTURED
DVI_ITEM_CODE -
SDI,    F,      L_DEVCHAR,      BITVAL, DEV$V_SDI,      1,      UCB,    ANY

;NAME,  SPECIAL,SOURCE,         DTYPE, BITPOS,          OUTLEN, STRUCT, DEVTYP
;

; SEQUENTIAL BLOCK-ORIENTED DEVICE (I.E., MAGTAPE)
DVI_ITEM_CODE -
SQD,    F,      L_DEVCHAR,      BITVAL, DEV$V_SQD,      1,      UCB,    ANY

; DEVICE BEING SPOOLED
```

```
DVI_ITEM_CODE -
SPL,    F,      L_DEVCHAR,      BITVAL, DEV$V_SPL,    1,      UCB,    ANY
; DEVICE IS AN OPERATOR
DVI_ITEM_CODE -
OPR,    F,      L_DEVCHAR,      BITVAL, DEV$V_OPR,    1,      UCB,    ANY
; DISK CONTAINS RCT (DEC STANDARD 166 DISK)
DVI_ITEM_CODE -
RCT,    F,      L_DEVCHAR,      BITVAL, DEV$V_RCT,    1,      UCB,    ANY
; NETWORK DEVICE
DVI_ITEM_CODE -
NET,    F,      L_DEVCHAR,      BITVAL, DEV$V_NET,    1,      UCB,    ANY

;NAME,  SPECIAL,SOURCE,         DTYPE,  BITPOS,       OUTLEN, STRUCT, DEVTYP
;
; FILES-ORIENTED DEVICE (I.E., DISK AND MT)
DVI_ITEM_CODE -
FOD,    F,      L_DEVCHAR,      BITVAL, DEV$V_FOD,    1,      UCB,    ANY
; DEVICE IS DUAL PORTED
DVI_ITEM_CODE -
DUA,    F,      L_DEVCHAR,      BITVAL, DEV$V_DUA,    1,      UCB,    ANY
; DEVICE SHAREABLE
DVI_ITEM_CODE -
SHR,    F,      L_DEVCHAR,      BITVAL, DEV$V_SHR,    1,      UCB,    ANY
; DEVICE IS A GENERIC DEVICE
DVI_ITEM_CODE -
GEN,    F,      L_DEVCHAR,      BITVAL, DEV$V_GEN,    1,      UCB,    ANY
; DEVICE AVAILABLE FOR USE
DVI_ITEM_CODE -
AVL,    F,      L_DEVCHAR,      BITVAL, DEV$V_AVL,    1,      UCB,    ANY

;NAME,  SPECIAL,SOURCE,         DTYPE,  BITPOS,       OUTLEN, STRUCT, DEVTYP
;
; DEVICE IS MOUNTED
DVI_ITEM_CODE -
MNT,    F,      L_DEVCHAR,      BITVAL, DEV$V_MNT,    1,      UCB,    ANY
; DEVICE IS A MAILBOX
DVI_ITEM_CODE -
MBX,    F,      L_DEVCHAR,      BITVAL, DEV$V_MBX,    1,      UCB,    ANY
; DEVICE MARKED FOR DISMOUNT
DVI_ITEM_CODE -
DMT,    F,      L_DEVCHAR,      BITVAL, DEV$V_DMT,    1,      UCB,    ANY
; DEVICE HAS ERROR LOGGING ENABLED
```

```
DVI_ITEM_CODE -
ELG,    F,      L_DEVCHAR,      BITVAL, DEV$V_ELG,      1,      UCB,    ANY
; DEVICE IS ALLOCATED
DVI_ITEM_CODE -
ALL,    F,      L_DEVCHAR,      BITVAL, DEV$V_ALL,      1,      UCB,    ANY

;NAME,  SPECIAL,SOURCE,         DTYPE,  BITPOS,         OUTLEN, STRUCT, DEVTYP
;
; DEVICE IS MOUNTED FOREIGN (I.E., NON-FILE STRUCTURED)
DVI_ITEM_CODE -
FOR,    F,      L_DEVCHAR,      BITVAL, DEV$V_FOR,      1,      UCB,    ANY
; DEVICE IS SOFTWARE WRITE LOCKED
DVI_ITEM_CODE -
SWL,    F,      L_DEVCHAR,      BITVAL, DEV$V_SWL,      1,      UCB,    ANY
; DEVICE CAPABLE OF PROVIDING INPUT
DVI_ITEM_CODE -
IDV,    F,      L_DEVCHAR,      BITVAL, DEV$V_IDV,      1,      UCB,    ANY
; DEVICE CAPABLE OF PROVIDING OUTPUT
DVI_ITEM_CODE -
ODV,    F,      L_DEVCHAR,      BITVAL, DEV$V_ODV,      1,      UCB,    ANY
; DEVICE ALLOWS RANDOM ACCESS
DVI_ITEM_CODE -
RND,    F,      L_DEVCHAR,      BITVAL, DEV$V_RND,      1,      UCB,    ANY

;NAME,  SPECIAL,SOURCE,         DTYPE,  BITPOS,         OUTLEN, STRUCT, DEVTYP
;
; DEVICE IS REALTIME IN NATURE
DVI_ITEM_CODE -
RTM,    F,      L_DEVCHAR,      BITVAL, DEV$V_RTM,      1,      UCB,    ANY
; DEVICE HAS READ CHECKING ENABLED
DVI_ITEM_CODE -
RCK,    F,      L_DEVCHAR,      BITVAL, DEV$V_RCK,      1,      UCB,    ANY
; DEVICE HAS WRITE CHECKING ENABLED
DVI_ITEM_CODE -
WCK,    F,      L_DEVCHAR,      BITVAL, DEV$V_WCK,      1,      UCB,    ANY
;
;   THE FOLLOWING CODES ARE THE INDIVIDUAL BITS OF THE DEVDEPEND LONGWORD
;   (AS DEFINED FOR TERMINALS: TTDEF IN STARDEFQZ.SDL)
;
;
;NAME,  SPECIAL,SOURCE,         DTYPE,  BITPOS,         OUTLEN, STRUCT, DEVTYP
;
```

```
DVI_ITEM_CODE -
TT_PASSALL, F,  L_DEVDEPEND,   BITVAL, TTSV_PASSALL,   1,     UCB,    ANY

DVI_ITEM_CODE -
TT_NOECHO, F,   L_DEVDEPEND,   BITVAL, TTSV_NOECHO,    1,     UCB,    ANY

DVI_ITEM_CODE -
TT_NOTYPEAHD, F, L_DEVDEPEND,  BITVAL, TTSV_NOTYPEAHD, 1,     UCB,    ANY

DVI_ITEM_CODE -
TT_ESCAPE, F,   L_DEVDEPEND,   BITVAL, TTSV_ESCAPE,    1,     UCB,    ANY

DVI_ITEM_CODE -
TT_HOSTSYNC, F, L_DEVDEPEND,   BITVAL, TTSV_HOSTSYNC,  1,     UCB,    ANY

DVI_ITEM_CODE -
TT_TTSYNC, F,   L_DEVDEPEND,   BITVAL, TTSV_TTSYNC,    1,     UCB,    ANY

DVI_ITEM_CODE -
TT_SCRIPT, F,   L_DEVDEPEND,   BITVAL, TTSV_SCRIPT,    1,     UCB,    ANY

;NAME,  SPECIAL,SOURCE,        DTYPE, BITPOS,        OUTLEN, STRUCT, DEVTYP
;

DVI_ITEM_CODE -
TT_LOWER, F,    L_DEVDEPEND,   BITVAL, TTSV_LOWER,     1,     UCB,    ANY

DVI_ITEM_CODE -
TT_MECHTAB, F,  L_DEVDEPEND,   BITVAL, TTSV_MECHTAB,   1,     UCB,    ANY

DVI_ITEM_CODE -
TT_WRAP, F,     L_DEVDEPEND,   BITVAL, TTSV_WRAP,      1,     UCB,    ANY

DVI_ITEM_CODE -
TT_CRFILL, F,   L_DEVDEPEND,   BITVAL, TTSV_CRFILL,    1,     UCB,    ANY

DVI_ITEM_CODE -
TT_LFFILL, F,   L_DEVDEPEND,   BITVAL, TTSV_LFFILL,    1,     UCB,    ANY

DVI_ITEM_CODE -
TT_SCOPE, F,    L_DEVDEPEND,   BITVAL, TTSV_SCOPE,     1,     UCB,    ANY

DVI_ITEM_CODE -
TT_REMOTE, F,   L_DEVDEPEND,   BITVAL, TTSV_REMOTE,    1,     UCB,    ANY

;NAME,  SPECIAL,SOURCE,        DTYPE, BITPOS,        OUTLEN, STRUCT, DEVTYP
;

DVI_ITEM_CODE -
TT_EIGHTBIT, F, L_DEVDEPEND,   BITVAL, TTSV_EIGHTBIT, 1,     UCB,    ANY

DVI_ITEM_CODE -
TT_MBXDSABL, F, L_DEVDEPEND,   BITVAL, TTSV_MBXDSABL, 1,     UCB,    ANY
```

```
DVI_ITEM_CODE -
TT_NOBRDCST, F, L_DEVDEPEND,    BITVAL, TT$V_NOBRDCST, 1,      UCB,    ANY

DVI_ITEM_CODE -
TT_READSYNC, F, L_DEVDEPEND,    BITVAL, TT$V_READSYNC, 1,      UCB,    ANY

DVI_ITEM_CODE -
TT_MECHFORM, F, L_DEVDEPEND,    BITVAL, TT$V_MECHFORM, 1,      UCB,    ANY

DVI_ITEM_CODE -
TT_HALFDUP, F, L_DEVDEPEND,     BITVAL, TT$V_HALFDUP,  1,      UCB,    ANY

;
;NAME,  SPECIAL,SOURCE,         DTYPE, BITPOS,         OUTLEN, STRUCT, DEVTYP
;

DVI_ITEM_CODE -
TT_MODEM, F,    L_DEVDEPEND,    BITVAL, TT$V_MODEM,    1,      UCB,    ANY

DVI_ITEM_CODE -
TT_OPER, F,     L_DEVDEPEND,    BITVAL, TT$V_OPER,     1,      UCB,    ANY

DVI_ITEM_CODE -
TT_PAGE, F,     L_DEVDEPEND+3,  DECNUM, 0,             1,      UCB,    ANY

;
;   THE FOLLOWING CODES ARE THE INDIVIDUAL BITS OF THE DEVDEPND2 LONGWORD
;   (AS DEFINED FOR TERMINALS: TT2DEF IN STARDEFQZ.SDL)
;

;
;NAME,  SPECIAL,SOURCE,         DTYPE, BITPOS,         OUTLEN, STRUCT, DEVTYP
;

DVI_ITEM_CODE -
TT_LOCALECHO, F, L_DEVDEPND2,   BITVAL, TT2$V_LOCALECHO, 1,    UCB,    ANY

DVI_ITEM_CODE -
TT_AUTOBAUD, F, L_DEVDEPND2,    BITVAL, TT2$V_AUTOBAUD, 1,     UCB,    ANY

DVI_ITEM_CODE -
TT_HANGUP, F,   L_DEVDEPND2,    BITVAL, TT2$V_HANGUP,  1,      UCB,    ANY

DVI_ITEM_CODE -
TT_MODHANGUP, F, L_DEVDEPND2,   BITVAL, TT2$V_MODHANGUP, 1,    UCB,    ANY

DVI_ITEM_CODE -
TT_BRDCSTMBX, F, L_DEVDEPND2,   BITVAL, TT2$V_BRDCSTMBX, 1,    UCB,    ANY

DVI_ITEM_CODE -
TT_DMA, F,      L_DEVDEPND2,    BITVAL, TT2$V_DMA,     1,      UCB,    ANY

;
;NAME,  SPECIAL,SOURCE,         DTYPE, BITPOS,         OUTLEN, STRUCT, DEVTYP
;
```

```
DVI_ITEM_CODE -
TT_ALTYPEAHD, F, L_DEVDEPND2,   BITVAL, TT2$V_ALTYPEAHD, 1,      UCB,     ANY

DVI_ITEM_CODE -
TT_SETSPEED, F, L_DEVDEPND2,    BITVAL, TT2$V_SETSPEED, 1,       UCB,     ANY

; TEMP DEFINITIONS FOR DCL SPAWN

DVI_ITEM_CODE -
TT_DCL_MAILBX, F, L_DEVDEPND2,  BITVAL, TT2$V_DCL_MAILBX, 1,     UCB,     ANY

DVI_ITEM_CODE -
TT_EDITING, F, L_DEVDEPND2,     BITVAL, TT2$V_EDITING,  1,       UCB,     ANY

DVI_ITEM_CODE -
TT_INSERT, F,   L_DEVDEPND2,    BITVAL, TT2$V_INSERT,   1,       UCB,     ANY

DVI_ITEM_CODE -
TT_FALLBACK, F, L_DEVDEPND2,    BITVAL, TT2$V_FALLBACK, 1,       UCB,     ANY

DVI_ITEM_CODE -
TT_DIALUP, F,   L_DEVDEPND2,    BITVAL, TT2$V_DIALUP,   1,       UCB,     ANY

DVI_ITEM_CODE -
TT_SECURE, F,   L_DEVDEPND2,    BITVAL, TT2$V_SECURE,   1,       UCB,     ANY

DVI_ITEM_CODE -
TT_DISCONNECT, F, L_DEVDEPND2,  BITVAL, TT2$V_DISCONNECT, 1,     UCB,     ANY

DVI_ITEM_CODE -
TT_PASTHRU, F, L_DEVDEPND2,     BITVAL, TT2$V_PASTHRU,  1,       UCB,     ANY

DVI_ITEM_CODE -
TT_SIXEL, F,    L_DEVDEPND2,    BITVAL, TT2$V_SIXEL,    1,       UCB,     ANY

DVI_ITEM_CODE -
TT_DRCS, F,     L_DEVDEPND2,    BITVAL, TT2$V_DRCS,     1,       UCB,     ANY

DVI_ITEM_CODE -
TT_PRINTER, F, L_DEVDEPND2,     BITVAL, TT2$V_PRINTER,  1,       UCB,     ANY

DVI_ITEM_CODE -
TT_APP_KEYPAD, F, L_DEVDEPND2,  BITVAL, TT2$V_APP_KEYPAD, 1,     UCB,     ANY

;
;NAME,  SPECIAL,SOURCE,          DTYPE, BITPOS,         OUTLEN, STRUCT, DEVTYP
;

DVI_ITEM_CODE -
TT_SYSPWD, F,   L_DEVDEPND2,     BITVAL, TT2$V_SYSPWD,  1,       UCB,     ANY

DVI_ITEM_CODE -
TT_ANSICRT, F, L_DEVDEPND2,     BITVAL, TT2$V_ANSICRT, 1,       UCB,     ANY

DVI_ITEM_CODE -
TT_REGIS, F,    L_DEVDEPND2,    BITVAL, TT2$V_REGIS,   1,       UCB,     ANY
```

```
DVI_ITEM_CODE -
TT_BLOCK, F,     L_DEVDEPND2,    BITVAL, TT2$V_BLOCK,    1,      UCB,    ANY

DVI_ITEM_CODE -
TT_AVO, F,       L_DEVDEPND2,    BITVAL, TT2$V_AVO,      1,      UCB,    ANY

DVI_ITEM_CODE -
TT_EDIT, F,      L_DEVDEPND2,    BITVAL, TT2$V_EDIT,     1,      UCB,    ANY

DVI_ITEM_CODE -
TT_DECCRT, F,    L_DEVDEPND2,    BITVAL, TT2$V_DECCRT,   1,      UCB,    ANY


;NAME,  SPECIAL,SOURCE,          DTYPE, BITPOS,          OUTLEN, STRUCT, DEVTYP
;

DVI_ITEM_CODE -
STS,    F,       L_STS,          BITVEC, 0,              4,      UCB,    ANY


;NAME,  SPECIAL,SOURCE,          DTYPE, BITPOS,          OUTLEN, STRUCT, DEVTYP
;

DVI_ITEM_CODE -
DEVSTS, F,       W_DEVSTS,       BITVEC, 0,              2,      UCB,    ANY

; DEVCHAR2 - Device characteristics second longword
DVI_ITEM_CODE -
DEVCHAR2, F,     L_DEVCHAR2,     BITVEC, 0,              4,      UCB,    ANY

; FULLDEVNAM - Fully qualified device name
DVI_ITEM_CODE -
FULLDEVNAM, T, FULLDEVNAM,       PADSTR, 0,              16,     UCB,    ANY

; LOCKID - cluster-wide lock id
DVI_ITEM_CODE -
LOCKID, F,       L_LOCKID,       HEXNUM, 0,              4,      UCB,    ANY

; ALLDEVNAM - Allocation class plus device name
DVI_ITEM_CODE -
ALLDEVNAM, T,    ALLDEVNAM,      PADSTR, 0,              16,     UCB,    ANY

; VOLSETMEM - Volume set member
DVI_ITEM_CODE -
VOLSETMEM, T,    VOLSETMEM,      BITVAL, 0,              1,      VCB,    DISK

; DEVLOCKNAM - Lock name for any device
DVI_ITEM_CODE -
DEVLOCKNAM, T, DEVLOCKNAM,       HEXSTR, 0,              16,     VCB,    ANY


; Item codes for dual-path and shadow-set support
;
```

```
DVITABLE.MAR;1                              16-SEP-1984 17:07:40.28  Page  15


;NAME,            SPECIAL, SOURCE,          DTYPE, BITPOS,  OUTLEN, STRUCT, DEVTYP
;

; ALLOCLASS - Allocation class of host(s)
DVI_ITEM_CODE -
ALLOCLASS,        F, L_ALLOCLS,            DECNUM, 0,       4,      DDB,    ANY

; ALT_HOST_AVAIL - Host for alternate path is active
DVI_ITEM_CODE -
ALT_HOST_AVAIL,   T, ALT_HOST_AVAIL,       BITVAL, 0,       1,      UCB,    ANY

; ALT_HOST_NAME - Name of host for alternate path
DVI_ITEM_CODE -
ALT_HOST_NAME,    T, ALT_HOST_NAME,        PADSTR, 0,       16,     UCB,    ANY

; ALT_HOST_TYPE - Type of host for alternate path
DVI_ITEM_CODE -
ALT_HOST_TYPE,    T, ALT_HOST_TYPE,        PADSTR, 0,       16,     UCB,    ANY

; HOST_AVAIL - Primary host is active
DVI_ITEM_CODE -
HOST_AVAIL,       T, HOST_AVAIL,           BITVAL, 0,       1,      UCB,    ANY

; HOST_COUNT - Number of paths to the device
DVI_ITEM_CODE -
HOST_COUNT,       T, HOST_COUNT,           DECNUM, 0,       4,      UCB,    ANY

; HOST_NAME - Name of host serving the primary path
DVI_ITEM_CODE -
HOST_NAME,        T, HOST_NAME,            PADSTR, 0,       16,     UCB,    ANY

; HOST_TYPE - Type of primary HOST {today one of ''V780'', ''V750'' or ''HS50''}
DVI_ITEM_CODE -
HOST_TYPE,        T, HOST_TYPE,            PADSTR, 0,       16,     UCB,    ANY

; REMOTE_DEVICE - Device is not connected to local node
DVI_ITEM_CODE -
REMOTE_DEVICE,    T, REMOTE_DEVICE,        BITVAL, 0,       1,      UCB,    ANY

; SERVED_DEVICE - Device is served to the cluster
DVI_ITEM_CODE -
SERVED_DEVICE,    F, L_DEVCHAR2,           BITVAL, DEV$V_SRV, 1,    UCB,    ANY


;NAME,            SPECIAL, SOURCE,          DTYPE, BITPOS,  OUTLEN, STRUCT, DEVTYP
;

; SHDW_CATCHUP_COPYING - Catchup copy in progress
DVI_ITEM_CODE =
SHDW_CATCHUP_COPYING, T, SHDW_CATCHUP_COPYING, BITVAL, 0, 1,       UCB,    ANY

; SHDW_MASTER - Device is master name for shadow set
DVI_ITEM_CODE -
SHDW_MASTER,      T, SHDW_MASTER,          BITVAL, 0,       1,      UCB,    ANY
```

```
; SHDW_MASTER_NAME - Name of the "virtual" master device for a shadow set
DVI_ITEM_CODE -
SHDW_MASTER_NAME, T, SHDW_MASTER_NAME,  PADSTR, 0,      16,       UCB,    ANY

; SHDW_MEMBER - Device is one of the volumes making a shadow set
DVI_ITEM_CODE -
SHDW_MEMBER,    F, L_DEVCHAR2,           BITVAL, DEV$V_SSM, 1,    UCB,    ANY

; SHDW_MERGE_COPYING - Merge copy in progress
DVI_ITEM_CODE -
SHDW_MERGE_COPYING, T, SHDW_MERGE_COPYING, BITVAL, 0, 1,         UCB,    ANY

; SHDW_NEXT_MBR_NAME - Name of the next device in shadow set
DVI_ITEM_CODE -
SHDW_NEXT_MBR_NAME, T, SHDW_NEXT_MBR_NAME, PADSTR, 0,   16,       UCB,    ANY

;
; Item code for virtual terminal physical device name
;

;
;NAME,     SPECIAL, SOURCE,                DTYPE,  BITPOS, OUTLEN, STRUCT, DEVTYP
;

; TT_PHYDEVNAM - Physical Device Name String
DVI_ITEM_CODE -
TT_PHYDEVNAM,    T, TT_PHYDEVNAM,         PADSTR, 0,      4,       UCB,    ANY

DVI_ITEM_CODE -
TT_DECCRT2, F, L_DEVDEPND2,      BITVAL, TT2$V_DECCRT2, 1,        UCB,    ANY

; MEDIA_ID name and type items
DVI_ITEM_CODE -
MEDIA_NAME, T, L_MEDIA_ID,      PADSTR, 0, 4, UCB,    ANY

DVI_ITEM_CODE -
MEDIA_TYPE, T, L_MEDIA_ID,      PADSTR, 0, 4, UCB,    ANY

DVI_ITEM_CODE -
MEDIA_ID,   F, L_MEDIA_ID,      BITVEC, 0, 4, UCB,    ANY

;
; Add several spare codes for shadow volumes - If these are never needed
; or used, the codes can be reused in a future release.
;
DVI_ITEM_CODE -
SHDW_SPARE_BIT_1, T, SHDW_SPARE_BIT_1, BITVAL, 0, 1,    UCB,    ANY
DVI_ITEM_CODE =
SHDW_SPARE_BIT_2, T, SHDW_SPARE_BIT_2, BITVAL, 0, 1,    UCB,    ANY
DVI_ITEM_CODE =
SHDW_SPARE_STRING_1, T, SHDW_SPARE_STRING_1, PADSTR, 0, 64,    UCB,    ANY
DVI_ITEM_CODE -
SHDW_SPARE_STRING_2, T, SHDW_SPARE_STRING_2, PADSTR, 0, 64,    UCB,    ANY
DVI_ITEM_CODE -
SHDW_SPARE_INTEGER_1, T, SHDW_SPARE_INTEGER_1, HEXNUM, 0, 4,    UCB,    ANY
```

```
DVI_ITEM_CODE -
SHDW_SPARE_INTEGER_2, T, SHDW_SPARE_INTEGER_2, HEXNUM, 0, 4,    UCB,    ANY

;
;****************************************************************
;****************************************************************
;
;        ADD NEW ITEMS IMMEDIATELY IN FRONT OF THIS COMMENT
;
;****************************************************************
;****************************************************************
;
        .ENDM   DVI_GENERATE_TABLE
```

CLIMAC
REQ

TPAMAC
REQ

UTLDEFB
B32

STARMISC
MAR

EODEFM
MAR

TPAMAR
MAR

CALBYNAME
LIS

SCRPROLOG
REQ

STARLET
SDL

SCRTCB
REQ

OFFSET
MAR

SYITABLE
MAR

STARLET
LIS

EODEFB
B32

JPITABLE
MAR

DVITABLE
MAR

UTLDEFM
MAR

B32MSG
LIS

SCRTERM
REQ

RSXLBLDF
MAR

C74MSG
LIS