NAME

dv2dt − convert a binary TeX DVI file to DTL text representation

SYNOPSIS

dv2dt *input-DVI-file output-DTL-file*

If the filenames are omitted, then *stdin* and *stdout* are assumed.

DESCRIPTION

dv2dt converts a binary TEX DVI file to an editable text file in DTL (*DVI Text Language*) format. The companion **dt2dv**(1) utility can convert the DTL file back to a binary DVI file.

DVI COMMAND DESCRIPTION

TEX DVI files contain a compact binary description of typeset pages, as a stream of operation code bytes, each immediately followed by zero or more parameter bytes. The format of DVI files is fully described in Donald E. Knuth, *TEX: The Program*, Addison-Wesley (1986), ISBN 0-201-13437-3, as well as in the **dvitype**(1) literate program source code.

For convenience, we provide a summary of DVI commands here. In the following list, operation code bytes are given as unsigned decimal values, followed by their symbolic names (not present in the DVI file), and a short description. A designation like *b[+n]* means that the operation code byte is followed by a parameter *b* which uses *n* bytes, and is signed. Without the plus sign, the parameter is unsigned. Signed integer parameter values are always represented in two's complement arithmetic, which is the system followed by most computers manufactured today, including all personal computers and workstations.

0 set_char_0	Set character 0 from current font.
. . .	
127 set_char_127	Set character 127 from current font.
128 set1 c[1]	Set 1-byte unsigned character (uchar) number *c*.
129 set2 c[2]	Set 2-byte uchar number *c*.
130 set3 c[3]	Set 3-byte uchar number *c*.
131 set4 c[+4]	Set 4-byte signed character (schar) number *c*.
132 set_rule a[+4] b[+4]	Set rule, height *a*, width *b*.
133 put1 c[1]	Put 1-byte uchar *c*.
134 put2 c[2]	Put 2-byte uchar *c*.
135 put3 c[3]	Put 3-byte uchar *c*.
136 put4 c[+4]	Put 4-byte schar *c*.
137 put_rule a[+4] b[+4]	Put rule, height *a*, width *b*.
138 nop	Do nothing.
139 bop c0[+4] . . . c9[+4] p[+4]	Beginning of page. The parameters *c0 . . . c9* are the TEX page counters, the contents of TEX count registers *\count0 . . . \count9*. The parameter

	p is the byte offset from the beginning of the DVI file of the previous *bop* operation code byte. The first such command in the file has $p = -1$.
140 eop	End of page.
141 push	Push (*h,v,w,x,y,z*) onto stack.
142 pop	Pop (*h,v,w,x,y,z*) from stack.
143 right1 b[+1]	Move right *b* units.
144 right2 b[+2]	Move right *b* units.
145 right3 b[+3]	Move right *b* units.
146 right4 b[+4]	Move right *b* units.
147 w0	Move right *w* units.
148 w1 b[+1]	Move right *b* units, and set $w = b$.
149 w2 b[+2]	Move right *b* units, and set $w = b$.
150 w3 b[+3]	Move right *b* units, and set $w = b$.
151 w4 b[+4]	Move right *b* units, and set $w = b$.
152 x0	Move right *x* units.
153 x1 b[+1]	Move right *b* units, and set $x = b$.
154 x2 b[+2]	Move right *b* units, and set $x = b$.
155 x3 b[+3]	Move right *b* units, and set $x = b$.
156 x4 b[+4]	Move right *b* units, and set $x = b$.
157 down1 a[+1]	Move down *a* units.
158 down2 a[+2]	Move down *a* units.
159 down3 a[+3]	Move down *a* units.
160 down4 a[+4]	Move down *a* units.
161 y0	Move right *y* units.
162 y1 a[+1]	Move right *a* units, and set $y = a$.
163 y2 a[+2]	Move right *a* units, and set $y = a$.
164 y3 a[+3]	Move right *a* units, and set $y = a$.
165 y4 a[+4]	Move right *a* units, and set $y = a$.
166 z0	Move right *z* units.
167 z1 a[+1]	Move right *a* units, and set $z = a$.
168 z2 a[+2]	Move right *a* units, and set $z = a$.
169 z3 a[+3]	Move right *a* units, and set $z = a$.
170 z4 a[+4]	Move right *a* units, and set $z = a$.

171 fnt_num_0 — Set current font number *(f) = 0*.

. . .

234 fnt_num_63 — Set *f = 63*.

235 fnt1 k[1] — Set *f = k*.

236 fnt2 k[2] — Set *f = k*.

237 fnt3 k[3] — Set *f = k*.

238 fnt4 k[+4] — Set *f = k*.

239 xxx1 k[1] x[k] — Special string *x* with *k* bytes.

240 xxx2 k[2] x[k] — Special string *x* with *k* bytes.

241 xxx3 k[3] x[k] — Special string *x* with *k* bytes.

242 xxx4 k[4] x[k] — Special string *x* with (unsigned) *k* bytes.

243 fnt_def1 k[1] c[4] s[4] d[4] a[1] l[1] n[a+l] — Define font *k*. The parameters are:

- *c* Checksum for TFM file.
- *s* Scale factor, in DVI units.
- *d* Design size, in DVI units.
- *a* Length of the ‘‘area’’ or directory.
- *l* Length of the font name.
- *n* Area and font name string(s).

244 fnt_def2 k[2] c[4] s[4] d[4] a[1] l[1] n[a+l] — Define font *k*.

245 fnt_def3 k[3] c[4] s[4] d[4] a[1] l[1] n[a+l] — Define font *k*.

246 fnt_def4 k[+4] c[4] s[4] d[4] a[1] l[1] n[a+l] — Define font *k*.

247 pre i[1] num[4] den[4] mag[4] k[1] x[k] — Begin preamble. The parameters are:

- *i* DVI format. Standard TEX has *ID = 2*, and TEX-XET has *ID = 3*.
- *num* Numerator of 100 nm / DVI unit.
- *den* Denominator of 100 nm / DVI unit.
- *mag* 1000 * magnification.
- *k* Comment length.
- *x* Comment string.

248 post p[4] num[4] den[4] mag[4] l[4] u[4] s[2] t[2] — Begin postamble. The parameters are:

p Pointer to final bop.

num, den, mag
Duplicates of values in preamble.

l Height-plus-depth of tallest page, in DVI units.

u Width of widest page, in DVI units.

s Maximum stack depth needed to process this DVI file.

t Total number of pages (*bop* commands) present.

249 post_post q[4] i[1] 223 . . . 223 End postamble. The parameters are:

q Byte offset from the beginning of the DVI file to the *post* command that started the postamble.

i DVI format ID, as in the preamble.

223 At least four *223* bytes.

250 Undefined.

. . .

255 Undefined.

DTL COMMAND DESCRIPTION

A DTL file contains one line per command, with a limit of 1024 characters per line. Each command contains a symbolic operation name, followed by zero or more parameter values. The parameter value descriptions are not repeated here; they can be found in the previous section.

variety <variety-name> This command specifies the name of the DTL file type; it has no DVI file equivalent.

(text) Series of set_char commands, for printable ASCII text.

\(Literal ASCII left parenthesis in (text).

\) Literal ASCII right parenthesis in (text).

\\ Literal ASCII backslash in (text).

\" Literal ASCII double quote in (text).

\XY Set_char for character with hexadecimal code XY, not in parentheses, but by itself for readability.

s1, s2, s2, s3 Set, with (1,2,3,4)-byte charcodes.

sr *set_rule*.

p1, p2, p2, p3	Put, with (1,2,3,4)-byte charcodes.
pr	*put_rule*.
nop	*nop* (do nothing).
bop	*bop* (beginning of page).
eop	*eop* (end of page).
[	Push.
]	Pop.
r1, r2, r3, r4	Right, with (1,2,3,4)-byte argument.
w0, w1, w2, w3, w4	As in DVI.
x0, x1, x2, x3, x4	As in DVI.
d1, d2, d3, d4	Down, with (1,2,3,4)-byte argument.
y0, y1, y2, y3, y4	As in DVI.
z0, z1, z2, z3, z4	As in DVI.
fn	*fnt_num* (set current font to font number in 0 to 63).
f1, f2, f3, f4	*fnt* (set current font to (1,2,3,4)-byte font number).
special	*xxx* (special commands with (1,2,3,4)-byte string length).
fd	*fnt_def* (assign a number to a named font).
pre	Preamble.
post	*post* (begin postamble).
post_post	*post_post* (end postamble).
opcode	Undefined DVI command (250 to 255).

SAMPLE DTL FILE

The following 2-line TEX file

```
Hello.
\bye
```

when processed with the commands

```
tex hello.tex
dv2dt hello.dvi hello.dtl
```

produces this DTL file:

```
variety sequences-6
pre 2 25400000 473628672 1000 27 ' TeX output 1995.03.02:2334'
bop 1 0 0 0 0 0 0 0 0 0 -1
[
d3 -917504
]
d4 42152922
[
d4 -41497562
[
r3 1310720
```

```
fd1 0 11374260171 655360 655360 0 5 '' 'cmr10'
fn0
(Hello.)
]
]
d3 1572864
[
r4 15229091
(1)
]
eop
post 42 25400000 473628672 1000 43725786 30785863 2 1
fd1 0 11374260171 655360 655360 0 5 'cmr10'
post_post 152 2 223 223 223 223
```

The command

```
dt2dv hello.dtl hello.dvi
```

will reconstruct the original DVI file.

SEE ALSO

dt2dv(1), **dvitype**(1), **tex**(1).

FILES

**.dvi* binary TEX DVI file.

**.dtl* text representation of a TEX DVI file in *DVI Text Language* format.

AUTHOR

dv2dt and **dt2dv**(1) were written by

Geoffrey Tobin
Department of Electronic Engineering
La Trobe University
Bundoora, Victoria 3083
Australia
Tel: +61 3 479 3736
FAX: +61 3 479 3025
Email: <G.Tobin@ee.latrobe.edu.au>

These manual pages were primarily written by

Nelson H. F. Beebe, Ph.D.
Center for Scientific Computing
Department of Mathematics
University of Utah
Salt Lake City, UT 84112
Tel: +1 801 581 5254
FAX: +1 801 581 4148
Email: <beebe@math.utah.edu>