

Graphics drivers for `pict2e`*

Rolf Niepraschk[†], Hubert Gäßlein[‡] and Josef Tkadlec[§]

2009/08/05

1 Driver files

This file implements some of the currently supported drivers for the new version of the `pict2e` package. If the driver you use is not in this list then a `.def` file may be distributed with the `pict2e` package, or may be distributed with the standard `LATEX` graphics bundle, or may be distributed with your driver.

If not, send us some details of the driver's `\special` syntax, and we will try to produce a suitable file.

Note that some of these files are for graphics drivers to which we have no access, so they are untested. Please send any corrections to the `latexbugs` address or directly to the authors.

1.1 Template

A template for a `pict2e` driver file.

`\pIIE@mode` This macro serves as an indicator to the `pict2e` package which mode the driver supports:

- `-1` inapt/incapable (default, already set in `pict2e`)
 - `0` standard `LATEX` only
 - `1` PostScript
 - `2` PDF
- (other values are reserved for future use)

Incapable drivers should not alter the default value given by the `pict2e` package, or set it explicitly to `-1`.

```
1 <*template>
2 \def\pIIE@mode{-1}
```

*This document corresponds to `p2e-drivers.dtx` v0.1s, dated 2009/08/05, documentation dated 2009/08/05.

[†]`Rolf.Niepraschk@ptb.de`

[‡]`HubertJG@open.mind.de`

[§]`j.tkadlec@email.cz`

`\pIIE@code` The `pict2e` package expects the driver file to define the `\pIIE@code` command in a suitable way.

This command should locally establish the standard PostScript/PDF coordinate system (i.e., a cartesian coordinate system with positive x-axis pointing right and positive y-axis pointing up, and with unit 1 bp = 1/72 in), albeit with the origin at `TEX`'s current point instead of the lower left corner of the page.

Furthermore, it should save and restore the graphics state (`gsave/grestore` in PostScript, `q/Q` in PDF.) This may be achieved by using appropriate `\special` (or `\pdfliteral`, respectively) commands.

Moreover, this command should preserve (i.e., it should not change) the current colour as defined by the user via the commands of the `color` package from the graphics bundle.

Thus, the `\Gin@PS@restored` command that various `<driver>.def` files from the graphics bundle provide should usually come close to what is expected here.

```
3 \def\pIIE@code#1{  
4 </template>
```

1.2 dvips

A `pict2e` driver file for the `dvips` driver.

`\pIIE@mode` We are about to generate PostScript code.

```
5 <*dvips>  
6 \def\pIIE@mode{1}
```

`\pIIE@code` In this case the code inserted by the driver on behalf of the `\Gin@PS@restored` command performs a “0 `setgray`” operation, thus resetting any colour the user might have set by means of the `color` package. (See also `LATEX` problem report `graphics/3569`.) We therefore have to resort to the following kludge: As long as we output only simple picture objects, our operations are “atomic.” Hence, we won't need to set colours or gray shades within the PostScript code generated by `pict2e`; thus the offending `setgray` operator may as well be a no-op. To keep this redefinition local, we enclose the call to `\Gin@PS@restored` by a `save/restore` pair.

```
7 \def\pIIE@code#1{%  
8 \Gin@PS@raw{save /setgray { pop } def}%  
9 \Gin@PS@restored{#1}%  
10 \Gin@PS@raw{restore}%  
11 }  
12 </dvips>
```

1.3 pdftex

A `pict2e` driver file for the `pdftex` driver.

`\pIIE@mode` We are about to generate PDF code. (Only, if `pdfTEX` is actually generating PDF; otherwise nothing will be output.)

```

13 <*pdfTeX>
14 \begingroup
15   \@ifundefined{pdfoutput}{}{-%
16     \ifnum\pdfoutput>0\relax
17       \gdef\pIIE@mode{2}
18     \fi
19   }
20 \endgroup

```

`\pIIE@code` The save/restore operators are necessary here to prevent the change of the CTM (scaling and rotation operations) that `pict2e` inserts from propagating.

```

21 \ifcase\pIIE@mode\relax \or\or
22   \def\pIIE@code#1{\pdfliteral{ q #1 Q }}
23 \fi
24 </pdfTeX>

```

1.4 vtex

A `pict2e` driver file for the `vtex` driver.

`\pIIE@mode` With `VTEX`, we should use PostScript code also for PDF mode (Email from Michael Vulis, MicroPress).

```

25 <*vtex>
26 \begingroup
27   \@ifundefined{VTeXversion}{}{-%
28     \ifnum\OpMode>0\relax
29       \ifnum\OpMode<3\relax
30         \gdef\pIIE@mode{1}%
31       \fi
32     \fi
33   }
34 \endgroup

```

`\pIIE@code` Here `\Gin@PS@restored` suffices as provided by the graphics driver file `vtex.def`.

```

35 \ifcase\pIIE@mode\relax \or
36   \let\pIIE@code\Gin@PS@restored
37 \fi
38 </vtex>

```

1.5 dvipdfm

A `pict2e` driver file for the `dvipdfm` driver.

`\pIIE@mode` We are about to generate PDF code.

```

39 <*dvipdfm>
40 \def\pIIE@mode{2}

```

`\pIIE@code` This seems to be sufficient.

```

41 \def\pIIE@code#1{\special{pdf: content #1}}
42 </dvipdfm>

```

1.6 xetex

A pict2e driver file for the xetex driver (submitted by Apostolos Syropoulos).

```
\pIIE@mode We are about to generate PDF code.
43 <*xetex>
44 \def\pIIE@mode{2}

\pIIE@code
45 \def\pIIE@code#1{\special{pdf: literal q #1 Q}}
46 </xetex>
```

1.7 dvipdf

A pict2e driver file for the dvipdf driver (not yet implemented).

```
\pIIE@mode
47 <*dvipdf>
48 % \def\pIIE@mode{-1}

\pIIE@code This is the same as the definition for \Gin@PS@restored in dvipdf.def as defined
in drivers.dtx! Better use the higher-level macro instead of the \special?
49 % \def\pIIE@code#1{\special{" #1}} % \Gin@PS@restored{#1}
50 </dvipdf>
```

1.8 textures

A pict2e driver file for the textures driver (not yet implemented).

```
\pIIE@mode
51 <*textures>
52 % \def\pIIE@mode{-1}

\pIIE@code
53 % \def\pIIE@code#1{}
54 </textures>
```

1.9 dvipsone

A pict2e driver file for the dvipsone driver (not yet implemented).

```
\pIIE@mode
55 <*dvipsone>
56 % \def\pIIE@mode{-1}

\pIIE@code
57 % \def\pIIE@code#1{}
58 </dvipsone>
```

1.10 pctexps

A pict2e driver file for the pctexps driver (not yet implemented).

```
\pIIE@mode
59 ⟨*pctexps⟩
60 % \def\pIIE@mode{-1}

\pIIE@code
61 % \def\pIIE@code#1{}
62 ⟨/pctexps⟩
```

1.11 pctex32

A pict2e driver file for the pctex32 driver (not yet implemented).

```
\pIIE@mode
63 ⟨*pctex32⟩
64 % \def\pIIE@mode{-1}

\pIIE@code
65 % \def\pIIE@code#1{}
66 ⟨/pctex32⟩
```

2 A Sample Configuration File

This one is taken from `color.cfg` of the $\text{tE}_X/\text{T}_E\text{X}$ live distributions.

```
67 ⟨*cfg⟩
68 %% Select an appropriate default driver.
69 \begingroup
70 \chardef\x=0 %
71 % check pdfTeX
72 \@ifundefined{pdfoutput}{-}{%
73   \ifcase\pdfoutput
74   \else
75     \chardef\x=1 %
76   \fi
77 }%
78 % check VTeX
79 \@ifundefined{OpMode}{-}{%
80   \chardef\x=2 %
81 }%
82 % check XeTeX
83 \@ifundefined{XeTeXrevision}{-}{%
84   \chardef\x=3 %
85 }%
86 \expandafter\endgroup
87 \ifcase\x
```

```
88 % default case
89 \ExecuteOptions{dvips}%
90 \or
91 % pdfTeX is running in pdf mode
92 \ExecuteOptions{pdftex}%
93 \or
94 % VTeX is running
95 \ExecuteOptions{vtex}%
96 \else
97 % XeTeX is running
98 \ExecuteOptions{xetex}%
99 \fi
```

You can also specify other options to the `pict2e` package in the configuration file. For example, if you prefer PSTricks-like arrows, just uncomment the line below.

```
100 %% \ExecuteOptions{pstarrows}
101 </cfg>
```