

PSTricks - 2008
new macros and bugfixes for the basic packages
pstricks, pst-plot, pst-tree,
and pst-node

Herbert Voß*

June 26, 2008

*Herbert.Voss@pstricks.de;
thanks to Patrick Drechsler.

Contents

I	pstricks – package	4
1	pstricks.sty	4
1.1	Error messages	4
1.2	Optional arguments	4
2	pstricks.tex (1.25– 2008/06/26)	5
2.1	Makro <code>\psDEBUG</code>	5
2.2	Option <code>shift</code>	5
2.3	Option <code>gridfont</code>	6
2.4	<code>linejoin</code>	7
2.5	<code>linecap</code>	8
2.6	New arrowtype <code>D></code> and <code>D>D></code>	9
2.7	Fill style <code>penrose</code>	10
2.8	Transparent colors	11
2.8.1	Options <code>strokeopacity</code> and <code>opacity</code>	11
2.8.2	Fill style <code>shape</code>	13
2.9	<code>\addtopsstyle</code>	15
2.10	<code>\pscircle0A</code>	16
2.11	<code>\psTextFrame</code>	16
2.12	Special coordinates	18
2.13	Octogon-Symbol	18
2.14	Code changes	19
3	The PostScript header files	19
3.1	<code>pstricks.pro</code>	19
3.2	<code>pst-dots.pro</code>	19
II	pst-node – package	21
4	pst-node.tex (1.00– 2007/10/16)	21
4.1	Bugfix for <code>psmatrix</code>	21
4.2	New option <code>pcRef</code>	21
III	pst-plot – package	24

5	pst-plot.tex (1.01– 2008/01/26)	24
5.1	New options LineToXAxis and LineToYAxis	24
IV	pst-tree – package	26
6	pst-tree.tex (1.01– 2007/06/26)	26

Part I

pstricks – package

1 pstricks.sty

1.1 Error messages

- Loading the package `pstricks` by L^AT_EX will now write a message into the file list of file version and date for the file `pstricks.pro`.
- A frequently done error is choosing a file name for the document, which is already a name of one PStricks package, e.g. `pstricks.tex`. The error message in the log file was not really helpful. There is now an extended message (example for a document file called `pstricks.tex`):

```
! LaTeX Error: ‘pstricks.tex’ is a forbidden name for your document,
it is already a name of a package.
```

See the L^AT_EX manual or L^AT_EX Companion for explanation.

Type `H` <return> for immediate help.

...

```
1.13 \documentclass
      {article}
```

? H

Choose another name for your document

1.2 Optional arguments

`pstricks` supports transparent colors with Ghostscript’s `.setopacityalpha`, `.setblendmode`, and `.setshapealpha`. These functions are not known to V_TE_X or Adobes Distiller. The optional argument `vtex` disables transparencies and `distiller` overrides the Ghostscript functions with the ones from the Distiller.

2 pstricks.tex (1.25– 2008/06/26)

2.1 Makro \psDEBUG

pstricks.tex defines the option `PstDebug=0|1`, which can be used for debugging. The new macro `\psDEBUG` makes it easier to write some debugging information into the package files. The macro is only valid, if `PstDebug=1` is set, otherwise the macro does nothing.

`\psDEBUG[optional arg]{text}`

`\psDEBUG` writes the argument `text` into the log file. Without an optional argument the word `pstricks` is used. The following output of the log file

```
1 ...
2 <key:xticksiz>: setting ticksize to max
3 LaTeX Font Info: External font 'cmex10' loaded for size
4 (Font)          <7> on input line 26.
5 LaTeX Font Info: External font 'cmex10' loaded for size
6 (Font)          <5> on input line 26.
7 <pst@@hlabels>: xticksizC=0.0pt
8 ...
```

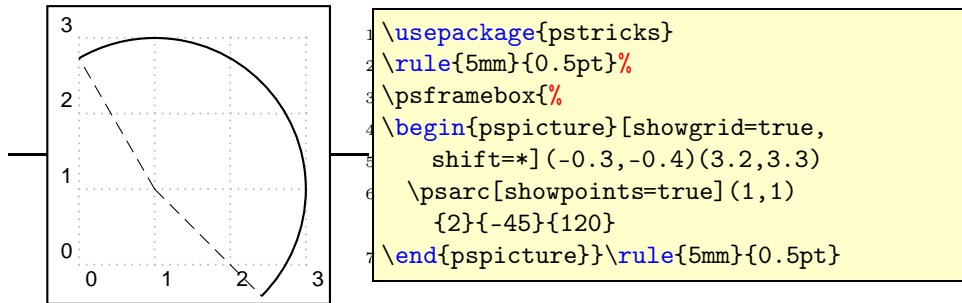
was possible with `\psset{PstDebug=1}`¹ and inside of `pstricks-add` with (only the first for example):

```
1 ...
2 \psDEBUG[key:ticksiz]{setting ticksize}
3 ...
```

2.2 Option shift

The optional argument `shift` can be used for a vertical alignment of the `pspicture` box. With `shift=*`, instead of a value or a length it is possible to center the `pspicture` box vertically to the baseline of the current line.

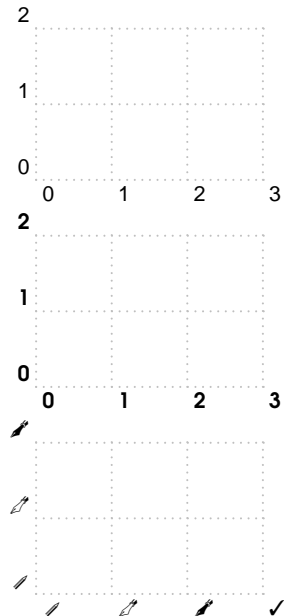
¹Can also be used locally for a macro when used as optional argument in the usual way.



2.3 Option gridfont

By default the grid labels were printed always in Helvetica. With the new keyword `gridfont` one can define another PostScript Font. Available are at least

Helvetica (default) - Helvetica-Narrow - Times-Roman - Courier - AvantGard - NewCenturySchlbk - Palatino-Roman - Bookman-Demi - ZapfDingbats - Symbol



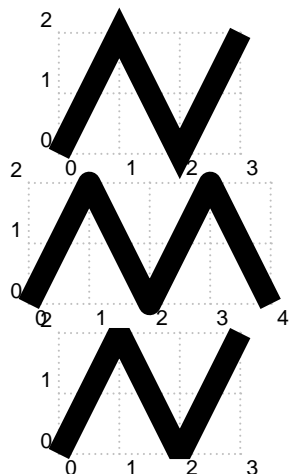
```

1 \usepackage{pstricks}
2 \begin{pspicture}[showgrid=true](3,2)
3 \end{pspicture}\[20pt]
4 \begin{pspicture}(3,2)
5   \psgrid[style=gridstyle,gridfont=
6     AvantGard-Demi]
7 \end{pspicture}\[20pt]
8 \begin{pspicture}(3,2)
9   \psgrid[style=gridstyle,gridfont=
10     ZapfDingbats]
11 \end{pspicture}

```

2.4 linejoin

Connecting lines can be done in several ways and is controlled on PS level by the `setlinejoin` command. With this version of PSTricks it is possible to controll this by an optional argument, called `linejoin`. It is preset to 0 and can take values of 0,1,2. Other values will have no effect.



```

1 \psset{linewidth=3mm,unit=0.8}
2 \begin{pspicture}[showgrid=true](3,2)
3   \psline(0,0)(1,2)(2,0)(3,2)
4 \end{pspicture}\\[10pt]
5 \begin{pspicture}[showgrid=true](4,2)
6   \psline[linejoin=1](0,0)(1,2)(2,0)(3,2)
7   (4,0)%
8 \end{pspicture}\\[10pt]
9 \begin{pspicture}[showgrid=true](3,2)
10  \psline[linejoin=2](0,0)(1,2)(2,0)(3,2)%
11 \end{pspicture}

```

2.5 linecap

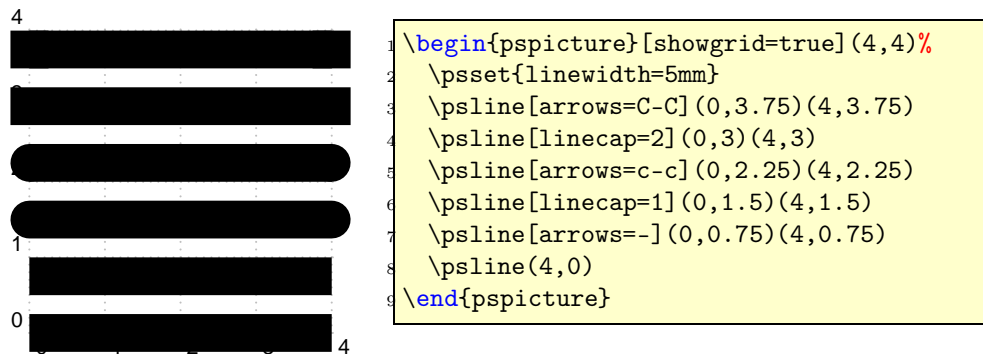
The value of `linecap` determines how the line ends are drawn:

0 lines are cut (default)

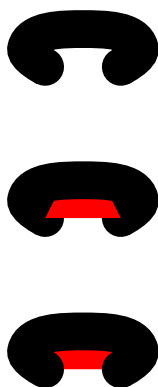
1 lines are ended by a filled semicircle of radius $0.5 \cdot \text{pslinewidth}$

2 lines are ended by a filled half square of radius $0.5 \cdot \text{pslinewidth}$

The following example shows that using `linecap` for lines is the same than using the arrow option.



Using this optional argument makes only sense in some special cases, because it is the same as the arrow type `c-c`. But the arrows are not part of the current path and filling an open curve with the `linecap` option is different to a curve using the `c-c` arrow.



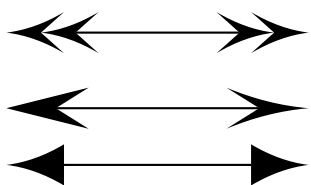
```

1 \psset{unit=5cm,linewidth=5mm}
2 \begin{pspicture}(-0.2,-0.6)(0.2,0.5)%
3 \def\curve{\pscurve(-.1,.1)(-.15,.15)(0,.2)(.15,.15)(.1,.1)}
4 \rput(0,.2){\psset{arrows=c-c}\curve}
5 \rput(0,-.2){%
6   \psset{fillstyle=solid,fillcolor=red,arrows=c-c}
7   \curve}
8 \rput(0,-.6){%
9   \psset{fillstyle=solid,fillcolor=red,linecap=1}
10  \curve}
11 \end{pspicture}

```


2.6 New arrowtype D> and D>D>

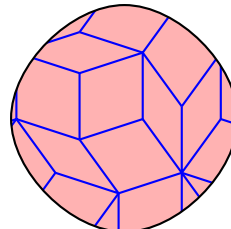
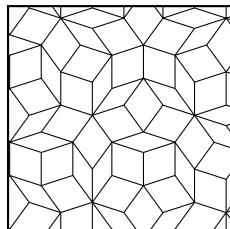
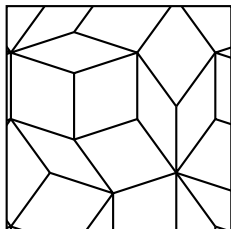
All arrows are drawn as polygons. The new arrow type D> or <D for the other way round, draws its lines as bezier curves, which looks nicer for big arrows.



```
1 \psset{arrowscale=5}  
2 \begin{pspicture}(4,2)  
3 \psline{<D<D-D>D>}(0,2)(4,2)  
4 \psline[arrows=<-D>,arrowlength=2](0,1)(4,1)  
5 \psline[arrowinset=0]{<D-D>}(0,0.25)(4,0.25)  
6 \end{pspicture}
```

2.7 Fill style penrose

The valid optional arguments are `penrose`, `penrose*`, and `hatchcolor`. The star version is only seen, if there is a fillcolor or a background different to white.



```
1 \begin{pspicture}(3,3)
2 \psframe[fillstyle=penrose](3,3)
3 \end{pspicture} \quad
4 \begin{pspicture}(3,3)
5 \psframe[fillstyle=penrose,psscale=0.5](3,3)
6 \end{pspicture} \quad
7 \begin{pspicture}(3,3)
8 \psccurve[fillstyle=penrose*,fillcolor=red!30,hatchcolor=blue](0,1.5)(1.5,3)
9 \end{pspicture}
```

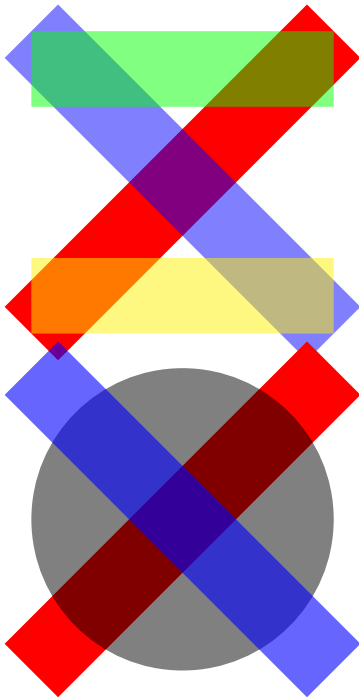
2.8 Transparent colors

The package `pstricks-add` already defined a fillstyle for transparency colors by using the Ghostscript's `blendmode`. It now moves into the main `pstricks` package, together with another possibility for creating transparent colors. Transparency is only seen with the PDF output (version 1.4 or greater), as nearly all PostScript viewer cannot show transparencies.

Loading the `pstricks` package with the option `vtex`, disables the transparency effects and everything works as before.

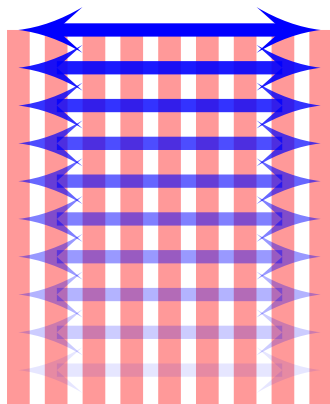
2.8.1 Options `strokeopacity` and `opacity`

For the existing fill style `solid` the new option `opacity` can be used to get also transparent colors. It is predefined by 1 (0...1), which is the old behaviour, no transparency. The option is valid only for PostScripts fill commands. Lines and curves can be transparent with setting the option `strokeopacity`, which can have a different value than the `opacity` option.



```
1 \begin{pspicture}[linewidth=1cm] (4,4)
2   \psline[linecolor=red] (0,0) (4,4)
3   \psline[linecolor=blue,strokeopacity=0.5] (0,4) (4,0)
4   \psline[linecolor=green,strokeopacity=0.5] (0,3.5)
5     (4,3.5)
6   \psline[linecolor=yellow,strokeopacity=0.5] (0,0.5)
7     (4,0.5)
8 \end{pspicture}
```

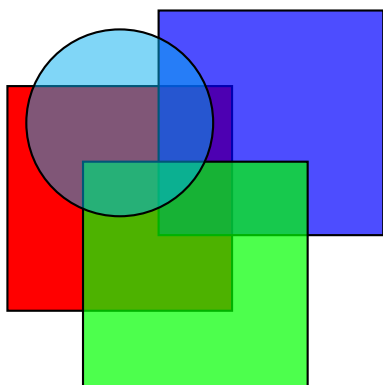
```
1 \begin{pspicture}[linewidth=1cm] (4,4)
2   \psline[linecolor=red] (0,0) (4,4)
3   \pscircle*[opacity=0.5] (2,2){2}
4   \psline[linecolor=blue,strokeopacity=0.6] (0,4) (4,0)
5 \end{pspicture}
```



```

1 \begin{pspicture}[linewidth=3mm](4,5.5)
2   \multido{\rA=0.0+0.5}{9}{%
3     \psline[linecolor=red!40](\rA,0)(\rA,5)}
4   \multido{\rA=0.0+0.5,\rB=0.0+0.1}{11}{%
5     \psline[arrows=<D-D>,linecolor=blue,
6       linewidth=5pt,arrowscale=1.5,
7       strokeopacity=\rB](0,\rA)(4,\rA)}
8 \end{pspicture}

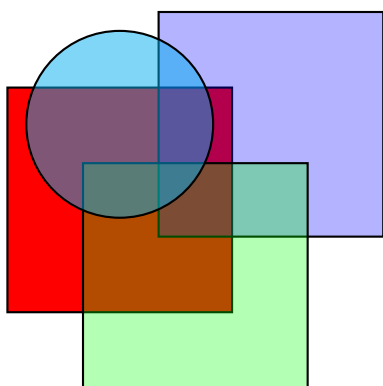
```



```

1 \begin{pspicture}(5,5)
2   \psset{fillstyle=solid}
3   \psframe[fillcolor=red](0,1)(3,4)
4   \psframe[fillcolor=blue,opacity=0.7](2,2)(5,5)
5   \psframe[fillcolor=green,opacity=0.7](1,0)(4,3)
6   \pscircle[fillcolor=cyan,
7     opacity=0.5](1.5,3.5){1.25}
8 \end{pspicture}

```



```

1 \begin{pspicture}(5,5)
2   \psset{fillstyle=solid}
3   \psframe[fillcolor=red](0,1)(3,4)
4   \psframe[fillcolor=blue,opacity=0.3](2,2)(5,5)
5   \psframe[fillcolor=green,opacity=0.3](1,0)(4,3)
6   \pscircle[fillcolor=cyan,
7     opacity=0.5](1.5,3.5){1.25}
8 \end{pspicture}

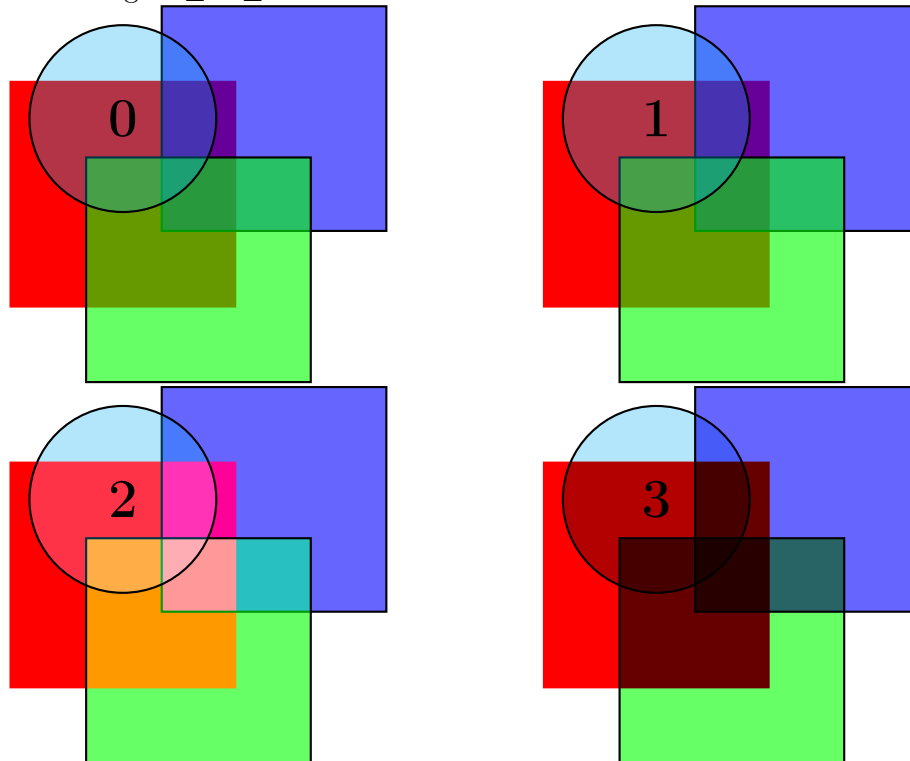
```

2.8.2 Fill style shape

There is now one more fill style for transparent colors: **shape** with using the **shapealpha** value and one of the possible blendmodes:

```
/Normal      ->0  
/Compatible ->1  
/Screen      ->2  
/Multiply    ->3
```

The fill style **solid** uses Ghostscript's **.setopacityalpha** function and the new style **shape** and the **blendmode** together with **.setshapealpha**. **shapealpha** is predefined with 0.6 and both alpha values can be chosen from the range $0 \leq \alpha \leq 1$.



```
1 \begin{pspicture}(5,5)% default blendmode  
2   \psframe*[linecolor=red](0,1)(3,4)  
3   \psframe[fillcolor=blue,fillstyle=shape](2,2)(5,5)  
4   \psframe[fillcolor=green,fillstyle=shape](1,0)(4,3)  
5   \pscircle[fillcolor=cyan,fillstyle=shape,  
6     shapealpha=0.3](1.5,3.5){1.25}  
7   \rput(1.5,3.5){\huge\textbf{0}}
```

```

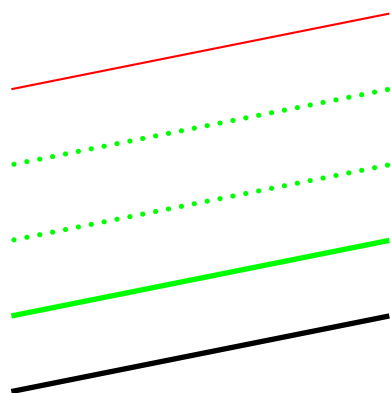
8 \end{pspicture}
9 \hfill
10 \begin{pspicture}(5,5)
11   \psset{blendmode=1}% type /Compatible
12   \psframe*[linecolor=red](0,1)(3,4)
13   \psframe[fillcolor=blue,fillstyle=shape](2,2)(5,5)
14   \psframe[fillcolor=green,fillstyle=shape](1,0)(4,3)
15   \pscircle[fillcolor=cyan,fillstyle=shape,
16     shapealpha=0.3](1.5,3.5){1.25}
17   \rput(1.5,3.5){\huge\textbf{1}}
18 \end{pspicture}
19
20 \begin{pspicture}(5,5)
21   \psset{blendmode=2}% type /Screen
22   \psframe*[linecolor=red](0,1)(3,4)
23   \psframe[fillcolor=blue,fillstyle=shape](2,2)(5,5)
24   \psframe[fillcolor=green,fillstyle=shape](1,0)(4,3)
25   \pscircle[fillcolor=cyan,fillstyle=shape,
26     shapealpha=0.3](1.5,3.5){1.25}
27   \rput(1.5,3.5){\huge\textbf{2}}
28 \end{pspicture}
29 \hfill
30 \begin{pspicture}(5,5)
31   \psset{blendmode=3}% type /Multiply
32   \psframe*[linecolor=red](0,1)(3,4)
33   \psframe[fillcolor=blue,fillstyle=shape](2,2)(5,5)
34   \psframe[fillcolor=green,fillstyle=shape](1,0)(4,3)
35   \pscircle[fillcolor=cyan,fillstyle=shape,
36     shapealpha=0.3](1.5,3.5){1.25}
37   \rput(1.5,3.5){\huge\textbf{3}}
38 \end{pspicture}

```

2.9 \addtopsstyle

`\addtopsstyle{style-name}{settings}`

This macro allows to add some more settings to an existing style. If the style is not defined, then `\addtopsstyle` behaves like the already defined `\newpsstyle` macro.

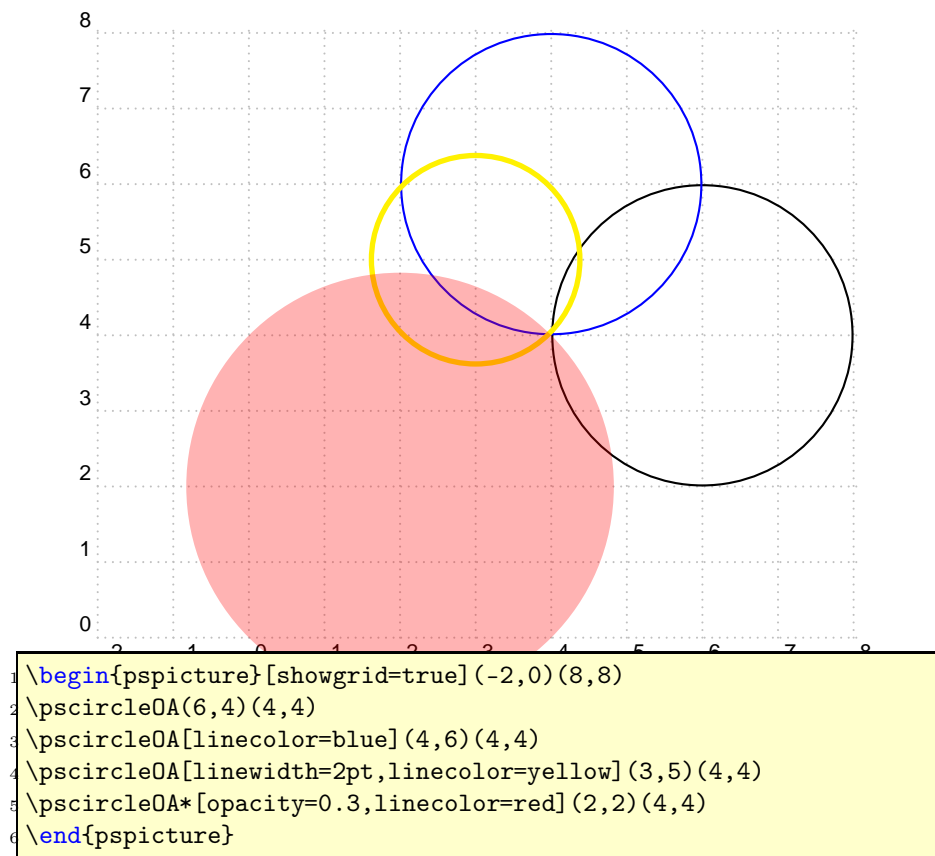


```
1 \newpsstyle{Fiber}{linewidth=2pt}
2 \begin{pspicture}(5,5)
3   \psline[style=Fiber](0,0)(5,1)
4   \addtopsstyle{Fiber}{linecolor=green}
5   \psline[style=Fiber](0,1)(5,2)
6   \addtopsstyle{Fiber}{linestyle=dotted}
7   \psline[style=Fiber](0,2)(5,3)
8   \addtopsstyle{Fiber}{}
9   \psline[style=Fiber](0,3)(5,4)
10  \addtopsstyle{Fibber}{linecolor=red}
11  \psline[style=Fibber](0,4)(5,5)
12 \end{pspicture}
```

2.10 `\pscircleOA`

`\pscircleOA[settings](x0,y0)(xA,yA)`

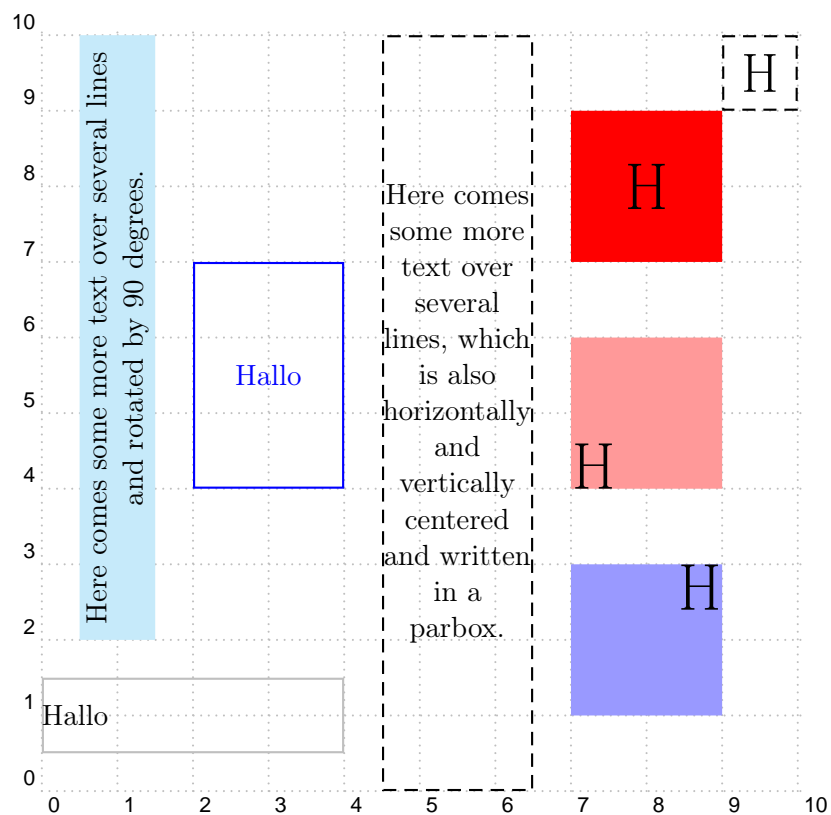
(x_0, y_0) is the center and (x_A, y_A) a given point of the circle. The radius is calculated by T_EX.



2.11 `\psTextFrame`

`\psTextFrame[settings](x1,y1)(x2,y2){Text}`

The *Text* cannot have a linebreak. In case it is needed, put the *Text* into a `minipage` or `\parbox`, as seen in the following example. The `ref`-option allows different placing and the `rot`-option allows the rotating of the *Text*. The macro itself first uses the `\psframe` and the `\rput` macro with calculated coordinates.



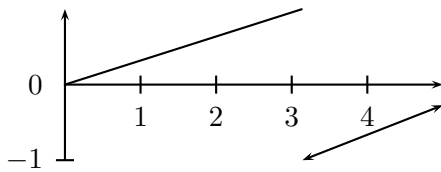
```

1 \begin{pspicture}[showgrid=true](0,-0.5)(10,10)
2 \psTextFrame[linecolor=lightgray,ref=1](0,0.5)(4,1.5){Hallo}
3 \psTextFrame[linecolor=blue](2,4)(4,7){\color{blue}Hallo}
4 \psTextFrame[linestyle=dashed](9,9)(10,10){\huge H}
5 \psTextFrame*[linecolor=red,linestyle=dashed](7,7)(9,9){\huge H}
6 \psTextFrame*[linecolor=red!40,ref=1B](7,4)(9,6){\huge H}
7 \psTextFrame*[linecolor=blue!40,ref=rt](7,1)(9,3){\huge H}
8 \psTextFrame[linestyle=dashed](4.5,0)(6.5,10){%
9 \parbox{2cm}{\centering Here comes some more text over several
10 lines, which is also horizontally and vertically centered and
11 written in a parbox.}}
12 \psTextFrame*[linecolor=cyan!20,rot=90](.5,2)(1.5,10){%
13 \parbox{8cm}{\centering Here comes some more text over several
14 lines and rotated by 90 degrees.}}
15 \end{pspicture}

```

2.12 Special coordinates

Additionally to the existing `!`-operator for PostScript coordinates, there is now a `*`-operator, which invokes the algebraic parser before the coordinates are passed to the default `!`-operator. The syntax is pretty easy: (`<value> {f(x)}`). In the following example the predefined value of `Pi` from `pstricks.pro` is used. The `x` value and the function must be enclosed in braces when they contain spaces, round braces or symbolic names, like `Pi` for the `x` value.

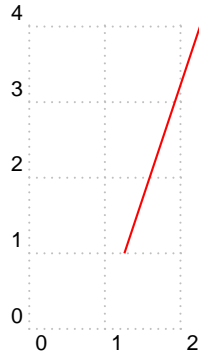


```

1 \SpecialCoord
2 \begin{pspicture}(0,-1)(5,1)
3   \psaxes{>}(0,0)(0,-1)(5,1)
4   \psline(0,0)(*{Pi} {sqrt(abs(cos(x)))})
5   \psline{<->)(*{Pi} {cos(x)})(*5 {sin(x)*cos(x)}
6 \end{pspicture}

```

For a macro definition of the algebraic function the `\string` command has to be used, otherwise \TeX expects the math mode in the following example.



```

1 \def\F{\string x^2 }
2 \begin{pspicture}[showgrid=true](2,4)
3   \psline[linecolor=red](*1 {\F} )(*2 {\F} )
4 \end{pspicture}

```

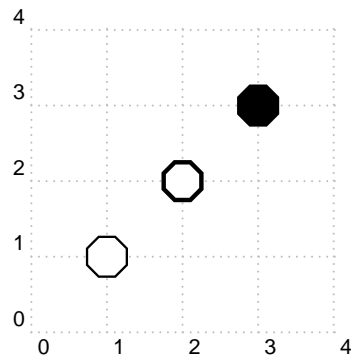
2.13 Octagon-Symbol

An octagon symbol was added.

```

\newpsfontdotH{Octagon}[1 0 0 1 0 0]{PSTricksDotFont}{(f)}{(g)}
\newpsfontdotH{BoldOctagon}[1 0 0 1 0 0]{PSTricksDotFont}{(F)}{(G)}
\newpsfontdot{SolidOctagon}[1 0 0 1 0 0]{PSTricksDotFont}{(g)}

```



```

1 \psset{dotstyle=4}
2 \begin{pspicture}[showgrid=true](4,4)
3 \psdot[dotstyle=Octagon](1,1)
4 \psdot[dotstyle=BoldOctagon](2,2)
5 \psdot[dotstyle=SolidOctagon](3,3)
6 \end{pspicture}

```

2.14 Code changes

```

% hv 2007-10-16 to fix the bug in pst-node with \[name=...]
\def\ps@ifnextchar#1#2#3{%
  \let\reserved@d= #1%
  \def\reserved@a{#2}\def\reserved@b{#3}%
  \futurelet\@let@token\ps@ifnch}
\def\ps@ifnch{%
  \ifx\@let@token\reserved@d \let\reserved@b\reserved@a \fi
  \reserved@b
}

```

3 The PostScript header files

3.1 pstricks.pro

```

/Pyth2 { % Pythagoras, xA yA xB yB
  3 -1 roll % xA xB yB yA
  sub % xA xB yB-yA
  3 1 roll % yB-yA xA xB
  sub % yB-yA xA-xB
  Pyth } def

```

This new PostScript function allows to calculate the distance between two points, given by their coordinates whereas the existing `/Pyth` does this for two values.

3.2 pst-dots.pro

`pst-dots.pro` is the file which defines the so called softfont `PSTRocksFont-Dot`, which collects all PSTricks symbols, which are not part of one of the standard PostScript fonts. The octagon symbol is defined as:

```

/OctagonPath {
  228 550 moveto 7 { -456 0 rlineto 45 rotate } repeat closepath
} def
/SolidOctagon { OctagonPath fill } def
/Octagon { OctagonPath .89 .89 scale OctagonPath eofill } def
/BoldOctagon { OctagonPath .79 .79 scale OctagonPath eofill } def
%
```

Part II

pst-node – package

4 pst-node.tex (1.00– 2007/10/16)

4.1 Bugfix for psmatrix

A long standing bug with `psmatrix` and using the `name` option is now fixed. The following works as expected:



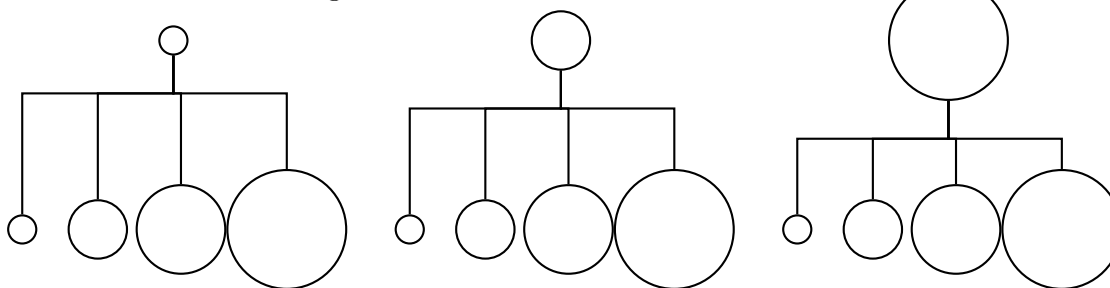
```
1 \begin{psmatrix}[rowsep=5mm]
2 [name=a]a\\
3 [name=b]b\\[1cm]
4 [name=c]c\\
5 \end{psmatrix}
6 \ncline{a}{b}
7 \ncarc{a}{c}
```

An optional argument after `\\` is now scanned in the correct way.

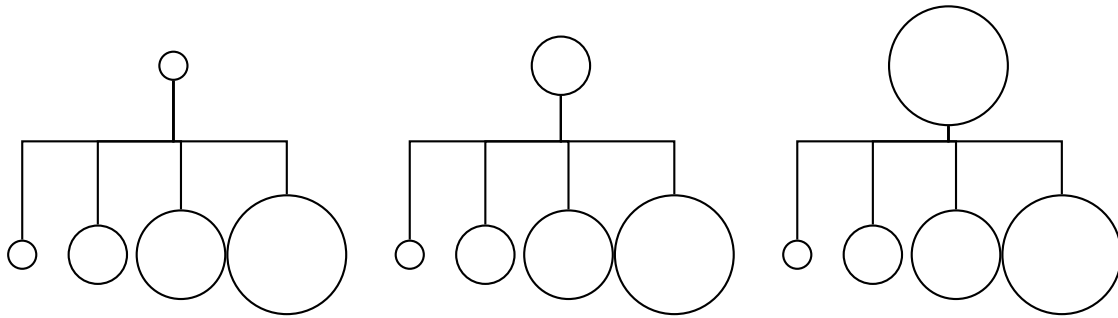
4.2 New option pcRef

There is a new option `pcRef` for the `\ncangles` connection. By default, the reference point for the `armA` option is the border of the node. This makes it difficult to get horizontally aligned lines for different node images. With `pcRef=true` the node center is the reference point and the connection is still drawn from the border of the node.

The first three images show the default behaviour:



The next three images display the influence of `pcRef=true`; the horizontal line for the three examples is on the same height:



```

1 \begin{pspicture}(5,4)
2   \cnode(2.5,3.5){0.2}{A}
3   \cnode(0.5,1){0.2}{B1}
4   \cnode(1.5,1){0.4}{B2}
5   \cnode(2.6,1){0.6}{B3}
6   \cnode(4,1){0.8}{B4}
7   \psset{angleB=90,angleA=-90,armA=1cm}
8   \ncangles[pcRef=true]{A}{B1}
9   \ncangles[pcRef=true]{A}{B2}
10  \ncangles[pcRef=true]{A}{B3}
11  \ncangles[pcRef=true]{A}{B4}
12 \end{pspicture}
13 %
14 \begin{pspicture}(5,4)
15   \cnode(2.5,3.5){0.4}{A}
16   \cnode(0.5,1){0.2}{B1}
17   \cnode(1.5,1){0.4}{B2}
18   \cnode(2.6,1){0.6}{B3}
19   \cnode(4,1){0.8}{B4}
20   \psset{angleB=90,angleA=-90,armA=1cm}
21   \ncangles[pcRef=true]{A}{B1}
22   \ncangles[pcRef=true]{A}{B2}
23   \ncangles[pcRef=true]{A}{B3}
24   \ncangles[pcRef=true]{A}{B4}
25 \end{pspicture}
26 %
27 \begin{pspicture}(5,4)
28   \cnode(2.5,3.5){0.8}{A}
29   \cnode(0.5,1){0.2}{B1}
30   \cnode(1.5,1){0.4}{B2}
31   \cnode(2.6,1){0.6}{B3}
32   \cnode(4,1){0.8}{B4}
33   \psset{angleB=90,angleA=-90,armA=1cm}
34   \ncangles[pcRef=true]{A}{B1}
35   \ncangles[pcRef=true]{A}{B2}

```

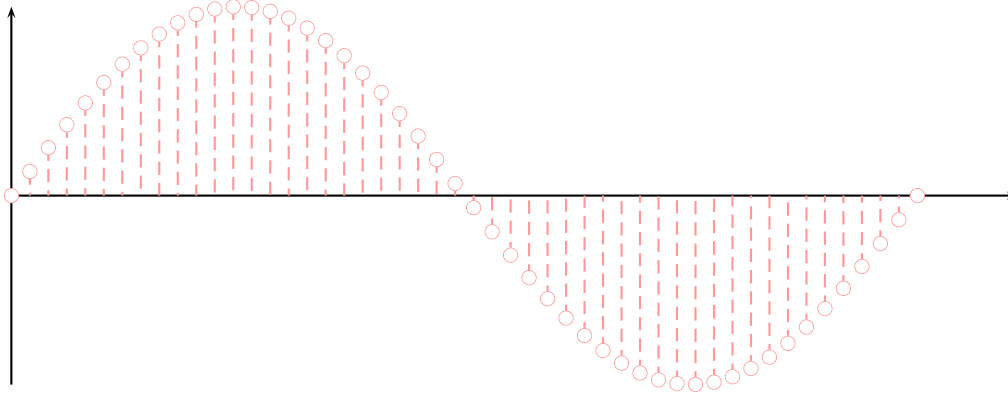
```
36 \ncangles[pcRef=true]{A}{B3}  
37 \ncangles[pcRef=true]{A}{B4}  
38 \end{pspicture}
```

Part III

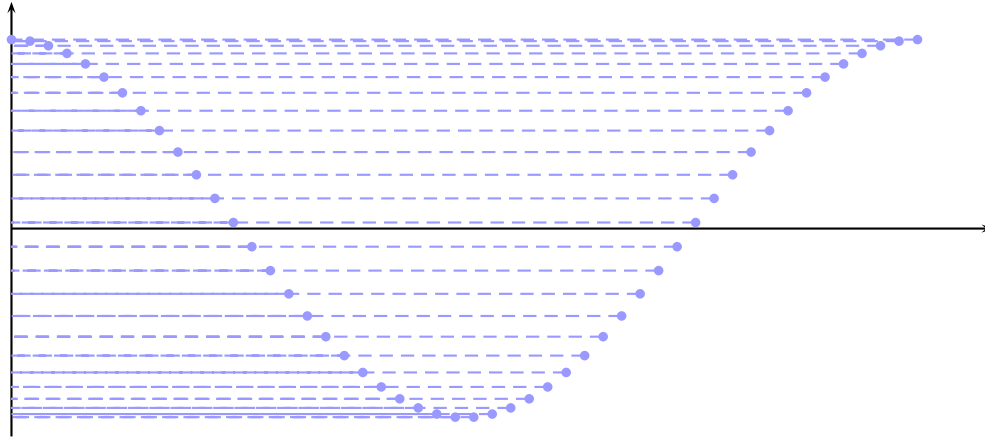
pst-plot – package

5 pst-plot.tex (1.01– 2008/01/26)

5.1 New options LineToXAxis and LineToYAxis



```
1 \psset{xunit=0.0333cm,yunit=2.5cm}
2 \begin{pspicture}(0,-1)(400,1)
3   \psline{->}(0,0)(400,0)
4   \psline{->}(0,-1)(0,1)
5   \psplot[plotstyle=LineToXAxis,linestyle=dashed,plotpoints=50,
6     linecolor=red!40,
7     showpoints=true,dotstyle=o,dotsize=0.2]{0}{360}{x sin}
8 \end{pspicture}
```

```

1 \psset{xunit=0.0333cm,yunit=2.5cm}
2 \begin{pspicture}(0,-1.2)(400,1.4)
3   \psline{->}(0,0)(390,0)
4   \psline{->}(0,-1.1)(0,1.2)
5   \psplot[plotstyle=LineToYAxis,linestyle=dashed,plotpoints=50,
6     linecolor=blue!40,
7     showpoints=true]{0}{360}{x cos}
8 \end{pspicture}

```

Part IV

pst-tree – package

6 pst-tree.tex (1.01– 2007/06/26)

There was a bug with `style=...` due to a missing `\use@par` in `\pstree@ii`.

References

- [1] Michel Goossens, Frank Mittelbach, Sebastian Rahtz, Denis Roegel, and Herbert Voß. *The L^AT_EX Graphics Companion*. Addison-Wesley Publishing Company, Reading, Mass., 2007.
- [2] Laura E. Jackson and Herbert Voß. Die Plot-Funktionen von `pst-plot`. *Die T_EXnische Komödie*, 2/02:27–34, June 2002.
- [3] Nikolai G. Kollock. *PostScript richtig eingesetzt: vom Konzept zum praktischen Einsatz*. IWT, Vaterstetten, 1989.
- [4] Manuel Luque. *Vue en 3D*. <http://members.aol.com/Mluque5130/vue3d16112002.zip>, 2002.
- [5] Herbert Voß. Die mathematischen Funktionen von Postscript. *Die T_EXnische Komödie*, 1/02:40–47, March 2002.
- [6] Herbert Voss. *PSTricks Support for pdf*. <http://PSTricks.de/pdf/pdfoutput.phtml>, 2002.
- [7] Herbert Voß. *L^AT_EX Referenz*. DANTE – Lehmanns, Heidelberg/Hamburg, 1. edition, 2007.
- [8] Herbert Voß. *PSTricks – Grafik für T_EX und L^AT_EX*. DANTE – Lehmanns, Heidelberg/Hamburg, 4. edition, 2007.
- [9] Michael Wiedmann and Peter Karp. *References for T_EX and Friends*. <http://www.miwie.org/tex-refs/>, 2003.
- [10] Timothy Van Zandt. *PSTricks - PostScript macros for Generic TeX*. <http://www.tug.org/application/PSTricks>, 1993.

Index

- π , 18
- `*`, 5
- `*-operator`, 18
- `.setblendmode`, 4
- `.setopacityalpha`, 4, 13
- `.setshapealpha`, 4, 13
- `/Pyth`, 19
- `<D`, 9
- `\`, 21
- `\addtopsstyle`, 15
- algebraic parser, 18
- `armA`, 21
- arrow, 8
- arrow type, 8
- AvantGard, 6
- blendmode, 11
- blendmode, 13
- Bookman-Demi, 6
- c-c, 8
- calculated coordinates, 16
- Courier, 6
- current path, 8
- D>, 9
- Dimension
 - `\pslinewidth`, 8
- Distiller, 4
- distiller, 4
- Environment
 - minipage, 16
 - psmatrix, 21
 - pspicture, 5
- error message, 4
- File
 - `psstricks.pro`, 4, 18
 - `psstricks.tex`, 4
- fill style, 13
- filling, 8
- Ghostscript, 4, 11, 13
- grid label, 6
- gridfont, 6
- Helvetica, 6
- Helvetica, 6
- Helvetica-Narrow, 6
- Keyword
 - `armA`, 21
 - blendmode, 13
 - gridfont, 6
 - linecap, 8
 - linejoin, 7
 - name, 21
 - opacity, 11
 - pcRef, 21
 - ref, 16
 - rot, 16
 - shape, 13
 - shapealpha, 13
 - shift, 5
 - strokeopacity, 11
- linecap, 8
- linejoin, 7
- Macro
 - `\`, 21
 - `\addtopsstyle`, 15
 - `\ncangles`, 21
 - `\newpsstyle`, 15
 - `\parbox`, 16

- `\psframe`, 16
- `\pstree@ii`, 26
- `\rput`, 16
- `\use@par`, 26
- minipage, 16
- name, 21
- `\ncangles`, 21
- NewCenturySchlbk, 6
- `\newpsstyle`, 15
- node connection, 21
- opacity, 11
- open curve, 8
- Package
 - `pstricks`, 4, 11
 - `pstricks-add`, 11
- Package option
 - `distiller`, 4
 - `vtex`, 4, 11
- Palatino-Roman, 6
- `\parbox`, 16
- `pcRef`, 21
- PDF, 11
- PostScript, 18
 - `.setblendmode`, 4
 - `.setopacityalpha`, 4, 13
 - `.setshapealpha`, 4, 13
 - `/Pyth`, 19
 - `setlinejoin`, 7
- PostScript Font, 6
- `\psframe`, 16
- `\pslinewidth`, 8
- `psmatrix`, 21
- `pspicture`, 5
- `\pstree@ii`, 26
- `pstricks`, 4, 11
- `pstricks-add`, 11
- `pstricks.pro`, 4, 18
- `pstricks.tex`, 4
- `ref`, 16
- `rot`, 16
- rotating, 16
- `\rput`, 16
- `setlinejoin`, 7
- shape, 13
- `shapealpha`, 13
- shift, 5
- solid, 11, 13
- `strokeopacity`, 11
- style, 15
- Symbol, 6
- Times-Roman, 6
- transparent colors, 4, 13
- transparent lines, 11
- transparency, 11
- `\use@par`, 26
- Value
 - `*`, 5
 - `<D`, 9
 - AvantGard, 6
 - Bookman-Demi, 6
 - `c-c`, 8
 - Courier, 6
 - `D>`, 9
 - Helvetica, 6
 - Helvetica-Narrow, 6
 - NewCenturySchlbk, 6
 - Palatino-Roman, 6
 - shape, 13
 - solid, 11, 13
 - Symbol, 6
 - Times-Roman, 6
 - ZapfDingbats, 6
- `m`, 4
- `vtex`, 4, 11
- ZapfDingbats, 6