# PSTricks

## News - 2009

**new macros and bugfixes for the basic packages `pstricks`, `pst-plot`, and `pst-node`**

December 9, 2013

2009

Package author(s):
**Herbert Voß**

# Contents

# Part I.
# pstricks – package

## 1. General

There exists a new document class pst-doc for writing PSTricks documentations, like this news document. It depends on the KOMA-Script document class scrartcl. pst-doc defines a lot of special macros to create a good index. Take one of the already existing package documentation and look into the source file. Then it will be easy to understand, how all these macros have to be used.

## 2. pstricks.sty

### 2.1. New optional argument

With the new optional argument 97 pstricks has the old behaviour of the version of 1997. The pspicture environment then has only a value as optional argument, which shifts the box in vertical direction. The new options shift and showgrid do not work in this case. The support of transparency colors is also disabled.

## 3. pstricks.tex (2.49– 2013/12/09)

### 3.1. Makro \next

To prevent problems with other packages which also define a \next macro, it is now renamed to \ps@next. This should be not important for the user, because it is used only by the base system itself.

### 3.2. Makro \psDEBUG

pstricks 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 [Options] {*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

```
...
<key:xticksize>: setting ticksize to max
LaTeX Font Info: External font ‘cmex10’ loaded for size
(Font)         <7> on input line 26.
LaTeX Font Info: External font ‘cmex10’ loaded for size
(Font)         <5> on input line 26.
<pst@@hlabels>: xticksizeC=0.0pt
```

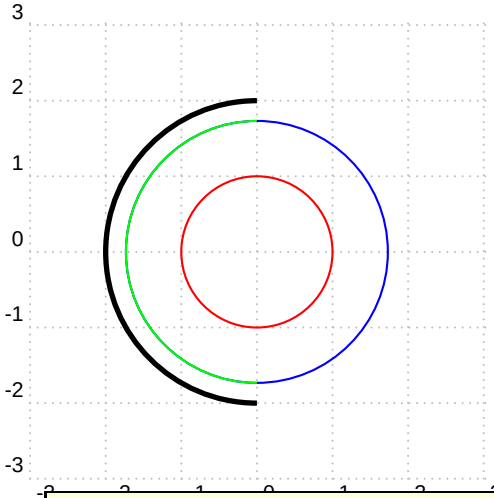
```
...
```

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

```
...
  \psDEBUG[key:ticksize]{setting ticksize}
...
```

## 3.3. Special Coordinates

The macros \SpecialCoor and the corresponding \NormalCoor can now be used for the radius of arcs and circles. With a preceeding ! character the length can be of any PostScript expression and will be taken with the unit pt in user coordinates:




```
\begin{pspicture}[showgrid=true](-3,-3)(3,3)
\pscircle[linecolor=red](0,0){1cm}
\SpecialCoor
\pscircle[linecolor=blue](0,0){!3 sqrt}
\NormalCoor
\psarc[linewidth=2pt](0,0){2cm}{90}{270}
\SpecialCoor
\psarc[linecolor=green](0,0){! 60 sin dup add }{90}{270}
\end{pspicture}
```

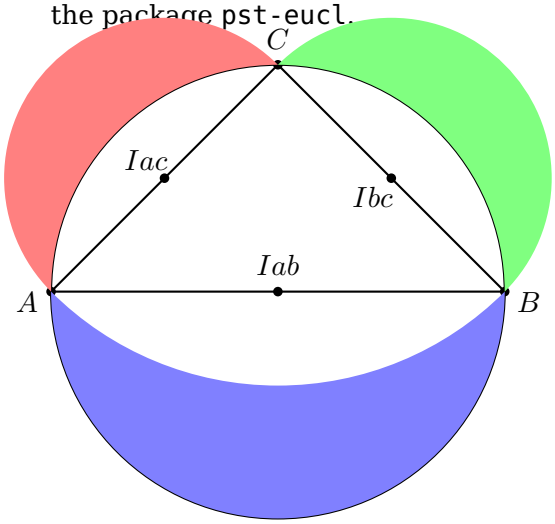
## 3.4. \psarcAB and \psarcnAB

\psarcAB [Options] {*arrows*} $(x_C,y_C)(x_A,y_A)(x_B,y_B)$
\psarcnAB [Options] {*arrows*} $(x_C,y_C)(x_A,y_A)(x_B,y_B)$

$(x_C,y_C)$ is the center and $(x_A,y_A)$ $(x_B,y_B)$ are two given points of the circle. The angles for the arc are calculated by TEX. The arc is drawn from the first to the second point. The radius is the distance from the center to the second point. The first point can have

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

another distance, it is only used for calculating the angle. The following examples need the package pst-eucl.

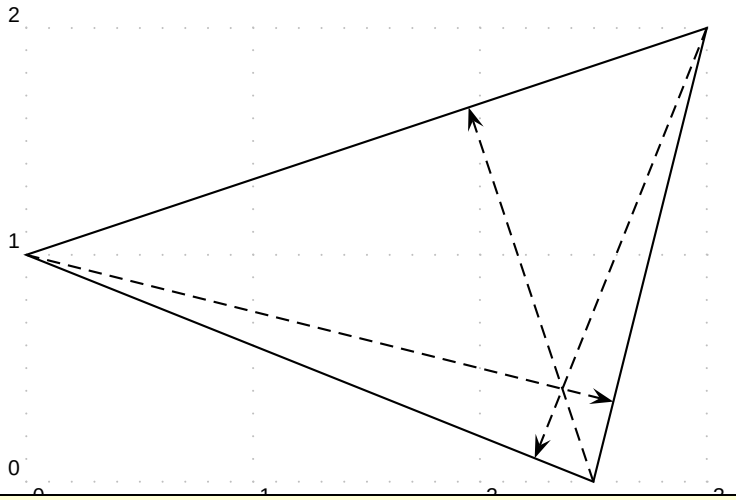



```
\begin{pspicture}(-3,-3)(3,3)
\pstTriangle[PosAngle={180,0,90}](-3,0){A}(3,0){B}(0,3){C}
\pstCircleOA{I}{A}
\pstMiddleAB{A}{B}{Iab}
\pscustom[linestyle=none,fillstyle=solid,fillcolor=blue!50]{%
  \psarcAB(C)(A)(B)
  \psarcnAB(Iab)(B)(A)}
\pstMiddleAB{A}{C}{Iac}
\pscustom[linestyle=none,fillstyle=solid,fillcolor=red!50]{%
  \psarcnAB(I)(A)(C)
  \psarcAB(Iac)(C)(A)}
\pstMiddleAB{B}{C}{Ibc}
\pscustom[linestyle=none,fillstyle=solid,fillcolor=green!50]{%
  \psarcAB(I)(B)(C)
  \psarcnAB(Ibc)(C)(B)}
\end{pspicture}
```

## 3.5. Makro \psPline

\psPline [Options] $(x_1,y_1)(x_2,y_2)(x_3,y_3)$

The first point P1 is a projection onto the line of $\overline{P_2P_3}$ and connected with drawing a line similiar to \psline. This macro has three mandatory arguments for the three points.




```
\psset{unit=3}
\begin{pspicture}[showgrid=true](3,2)
\pspolygon(0,1)(3,2)(2.5,0)
\psPline[linestyle=dashed,arrowscale=2]{->}(0,1)(3,2)(2.5,0)
\psPline[linestyle=dashed,arrowscale=2]{->}(3,2)(2.5,0)(0,1)
\psPline[linestyle=dashed,arrowscale=2]{->}(2.5,0)(0,1)(3,2)
\end{pspicture}
```

## 4. The PostScript header files

### 4.1. pstricks.pro

Only cosemetic changes to the code to make it more readable.

### 4.2. pst-node.pro

Only cosemetic changes to the code to make it more readable.

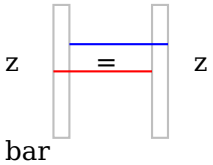
# Part II.
# pst-node – package

## 5. pst-node.tex (1.31– 2013/10/22)

There was a bug in `\fnode`. It is now fixed.

foo
z = z
bar

now with an aligned node
foo




bar

```
foo

   z \quad
     \pnode{testL1}
     \fnode[linecolor=lightgray,framesize=6pt 50pt](0,.5ex){eqL}
     \pnode(0,2ex){testL2}
     \quad
   =
     \quad%
     \pnode{testR1}
     \fnode[linecolor=lightgray,framesize=6pt 50pt](0,.5ex){eqR}
     \pnode(0,2ex){testR2}
     \quad
   z
 \ncline[linecolor=red]{testL1}{testR1}
 \ncline[linecolor=blue]{testL2}{testR2}

bar

\bigskip now with an aligned node
\psset{nodealign=true}

foo

   z \quad
     \pnode{testL1}
     \fnode[linecolor=lightgray,framesize=6pt 50pt](0,.5ex){eqL}
     \pnode(0,2ex){testL2}
     \quad
   =
     \quad%
     \pnode{testR1}
     \fnode[linecolor=lightgray,framesize=6pt 50pt](0,.5ex){eqR}
     \pnode(0,2ex){testR2}
     \quad
   z
 \ncline[linecolor=red]{testL1}{testR1}
 \ncline[linecolor=blue]{testL2}{testR2}

bar
```

# Part III.
# pst-plot – package

## 6. pst-plot.tex (1.57– 2013/12/01)

To be more consistent to other macro names there are now the names `\psparametricplot`, `\psfileplot`, `\pslistplot`, and `\psdataplot`. The old macros without the preceeding `ps` still work!

There is now a new `PostScript`boolean variable which can control loops. With setting `/ps@Exit true def` you can leave a loop for plotting a function with `\psplot` and `\psparametricplot`.

# Part IV.
# pst-tree – package

## 7. pst-tree.tex (1.12– 2011/02/28)

The package `pst-tree` now uses the advanced key handling from `xkeyval`. The reason why it moved from the base into the contrib sections, where all packages uses `xkeyval`.

# Index