

Cheat sheet for pst-optexp (v4.10)

General component parameters

labeloffset= $\langle num \rangle$
labelstyle= $\langle macros \rangle$
labelalign= $\langle refpoint \rangle$
labelangle= $\langle num \rangle$
labelref=relative, relgrav, global, absolute
label= $\langle offset \rangle$ [$\langle angle \rangle$] [$\langle refpoint \rangle$] [$\langle labelref \rangle$]
innerlabel=true
position= $\langle num \rangle$, start, end
abspos= $\langle num \rangle$, start, end
endbox=true, false
angle= $\langle pstyle \rangle$
rotateref= $\langle refpoint \rangle$
compshift= $\langle num \rangle$
compoffset= $\langle num \rangle$
innercompalign=rel, relative, abs, absolute
OptComp $\langle pstyle \rangle$
OptionalStyle $\langle pstyle \rangle$
VariableStyle $\langle pstyle \rangle$
addtoOptComp= $\langle list \rangle$
newOptComp= $\langle list \rangle$
optional=true, false

Free-ray components

$\backslash lens$ [$\langle opt \rangle$] ($\langle in \rangle$) ($\langle out \rangle$) { $\langle label \rangle$ }

lensheight= $\langle num \rangle$
lensradiusleft= $\langle num \rangle$
lensradiusright= $\langle num \rangle$
lensradius= $\langle left \rangle$ [$\langle right \rangle$]
lenswidth= $\langle num \rangle$
lens= $\langle radiusleft \rangle$ [$\langle radiusright \rangle$] [$\langle height \rangle$] [$\langle width \rangle$]
thicklens=true, false

$\backslash optplate$ [$\langle opt \rangle$] ($\langle in \rangle$) ($\langle out \rangle$) { $\langle label \rangle$ }

plateheight= $\langle num \rangle$
platelinewidth= $\langle num \rangle$ or $\langle dimen \rangle$

$\backslash optretplate$ [$\langle opt \rangle$] ($\langle in \rangle$) ($\langle out \rangle$) { $\langle label \rangle$ }

platewidth= $\langle num \rangle$
platesize= $\langle width \rangle$ $\langle height \rangle$

$\backslash pinhole$ [$\langle opt \rangle$] ($\langle in \rangle$) ($\langle out \rangle$) { $\langle label \rangle$ }

outerheight= $\langle num \rangle$
innerheight= $\langle num \rangle$
phlinewidth= $\langle num \rangle$ or $\langle dimen \rangle$
phwidth= $\langle num \rangle$

$\backslash optbox$ [$\langle opt \rangle$] ($\langle in \rangle$) ($\langle out \rangle$) { $\langle label \rangle$ }

optboxwidth= $\langle num \rangle$
optboxheight= $\langle num \rangle$
optboxsize= $\langle width \rangle$ $\langle height \rangle$

$\backslash optarrowcomp$ [$\langle opt \rangle$] ($\langle in \rangle$) ($\langle out \rangle$) { $\langle label \rangle$ }

arrowcompwidth= $\langle num \rangle$
arrowcompheight= $\langle num \rangle$
arrowcompsize= $\langle size \rangle$ or $\langle width \rangle$ $\langle height \rangle$
arrowcompangle= $\langle num \rangle$
arrowcompshape=rectangle, circle
ArrowCompStyle $\langle pstyle \rangle$

$\backslash optbarcomp$ [$\langle opt \rangle$] ($\langle in \rangle$) ($\langle out \rangle$) { $\langle label \rangle$ }

barcompwidth= $\langle num \rangle$
barcompheight= $\langle num \rangle$
barcompsize= $\langle size \rangle$ or $\langle width \rangle$ $\langle height \rangle$
barcompangle= $\langle num \rangle$
barcompshape=rectangle, circle
BarCompStyle $\langle pstyle \rangle$

$\backslash optsource$ [$\langle opt \rangle$] ($\langle in \rangle$) ($\langle out \rangle$) { $\langle label \rangle$ }

sourcewidth= $\langle num \rangle$
sourceheight= $\langle num \rangle$
sourcesize= $\langle width \rangle$ $\langle height \rangle$

$\backslash crystal$ [$\langle opt \rangle$] ($\langle in \rangle$) ($\langle out \rangle$) { $\langle label \rangle$ }

crystalwidth= $\langle num \rangle$
crystalheight= $\langle num \rangle$
crystalsize= $\langle width \rangle$ $\langle height \rangle$
caxislength= $\langle num \rangle$
caxisinv=true, false
voltage=true, false
lamp=true, false
CrystalCaxis $\langle pstyle \rangle$
CrystalLamp $\langle pstyle \rangle$

$\backslash optdiode$ [$\langle opt \rangle$] ($\langle in \rangle$) ($\langle out \rangle$) { $\langle label \rangle$ }

optdiodesize= $\langle num \rangle$

$\backslash doveprism$ [$\langle opt \rangle$] ($\langle in \rangle$) ($\langle out \rangle$) { $\langle label \rangle$ }

doveprismsize= $\langle num \rangle$ or $\langle width \rangle$ $\langle height \rangle$

$\backslash glanthompson$ [$\langle opt \rangle$] ($\langle in \rangle$) ($\langle out \rangle$) { $\langle label \rangle$ }

glanthompsonwidth= $\langle num \rangle$
glanthompsonheight= $\langle num \rangle$
glanthompsonsize= $\langle width \rangle$ $\langle height \rangle$
glanthompsongap= $\langle num \rangle$

$\backslash polarization$ [$\langle opt \rangle$] ($\langle in \rangle$) ($\langle out \rangle$) { $\langle label \rangle$ }

polsize= $\langle num \rangle$
poltype=parallel, perp, misc, lcirc, rcirc
Polarization $\langle pstyle \rangle$

$\backslash mirror$ [$\langle opt \rangle$] ($\langle in \rangle$) ($\langle center \rangle$) ($\langle out \rangle$) { $\langle label \rangle$ }

mirrorwidth= $\langle num \rangle$
mirrorlinewidth= $\langle num \rangle$ or $\langle dimen \rangle$
mirrorradius= $\langle radius \rangle$ [0]
mirrortype=plain, piezo, extended, semitrans
variable=true, false
mirrordepth= $\langle num \rangle$
ExtendedMirror $\langle pstyle \rangle$
PiezoMirror $\langle pstyle \rangle$
SemitransMirror $\langle pstyle \rangle$

$\backslash beamsplitter$ [$\langle opt \rangle$] ($\langle in \rangle$) ($\langle center \rangle$) ($\langle out \rangle$) { $\langle label \rangle$ }

bssize= $\langle num \rangle$
bsstyle=cube, plate

$\backslash optgrating$ [$\langle opt \rangle$] ($\langle in \rangle$) ($\langle center \rangle$) ($\langle out \rangle$) { $\langle label \rangle$ }

gratingwidth= $\langle num \rangle$
gratingheight= $\langle num \rangle$
gratingdepth= $\langle num \rangle$
gratingcount= $\langle int \rangle$
gratingtype=blazed, binary
gratingalign=t, top, c, center
reverse=true, false
gratinglinewidth= $\langle num \rangle$ or $\langle dimen \rangle$

$\backslash transmissiongrating$ [$\langle opt \rangle$] ($\langle in \rangle$) ($\langle center \rangle$) ($\langle out \rangle$) { $\langle label \rangle$ }

$\backslash optaom$ [$\langle options \rangle$] ($\langle in \rangle$) ($\langle trans \rangle$) ($\langle diff \rangle$) { $\langle label \rangle$ }

aomheight= $\langle num \rangle$
aomwidth= $\langle num \rangle$

```
aomsize=<width> <height>
aomgratingcount=<int>
aomalign=symmetric, straight
aomreflalign=perp, parallel
aomcomp=default, <macro>
diffractionorders=<int>
beamdiffractiionorder=<int>
```

```
\optprism[<opt>](<in>)(<center>)(<out>){<label>}
prismsize=<num>
prismangle=<num>
prismtype=transmittive, reflective
prismalign=auto, center
```

```
\rightangleprism[<opt>](<in>)(<center>)(<out>){<label>}
raprismsize=<num>
raprismalign=auto, center
```

```
\pentaprism[<opt>](<in>)(<center>)(<out>){<label>}
pentaprismsize=<num>
```

Fiber components

```
usefiberstyle=true, false
usewirestyle=true, false
```

```
\optfiber[<opt>](<in>)(<out>){<label>}
fiberloops=<int>
fiberloopradius=<num>
fiberloopsep=<num>
```

```
\optamp[<opt>](<in>)(<out>){<label>}
optampsize=<num> or <width> <height>
```

```
\optmzm[<opt>](<in>)(<out>){<label>}
optmzmsize=<num> or <width> <height>
```

```
\polcontrol[<opt>](<in>)(<out>){<label>}
polcontrolsize=<num>
polcontroltype=linear, triangle
```

```
\optisolator[<opt>](<in>)(<out>){<label>}
isolatorsizes=<num> or <width> <height>
IsolatorArrow <psstyle>
```

```
\optswitch[<opt>](<in>)(<out>){<label>}
switchsize=<num> or <width> <height>
```

```
switchstyle=opened, closed
```

```
\fiberdelayline[<opt>](<in>)(<out>){<label>}
fdlsize=<num> or <width> <height>
FdlArrow <psstyle>
```

```
\optfiberpolarizer[<opt>](<in>)(<out>){<label>}
fiberpolsize=<num> or <width> <height>
```

```
\optcirculator(<left>)(<right>)(<bottom>){<label>}
optcircsize=<num>
optcircangleA=<num>
optcircangleB=<num>
optcircangle=<num> <num>
OptCircArrow <psstyle>
```

```
\optcoupler(<tl>)(<bl>)(<tr>)(<br>){<label>}
\wdmcoupler(<tl>)(<bl>)(<r>){<label>}
\wdmsplitter(<l>)(<tr>)(<br>){<label>}
couplersize=<num> or <width> <height>
couplersep=<num>
couplertype=none, ellipse, rectangle, cross
coupleralign=t, top, b, bottom, c, center
VariableCoupler <psstyle>
```

```
\fiberbox(<in>)(<out>){<label>}
fiberboxwidth=<num>
fiberboxheight=<num>
fiberboxsize=<width> <height>
fiberboxsep=<num>
fiberboxsepout=<num>
fiberboxcount=<N>x<M>
```

Electrical components

```
\eleccoupler(<tl>)(<bl>)(<tr>)(<br>){<label>}
eleccouplersize=<size> or <width> <height>
eleccouplersep=<num>
eleccouplertype=standard, directional
eleccouplerinput=left, right
```

```
\elecsynthesizer(<in>)(<out>){<label>}
synthsize=<size> or <width> <height>
synthtype=sine, pulse, sawtooth, rectangle,
triangle, custom
```

```
synthshape=circle, rectangle
SynthStyle <psstyle>
```

```
\elecmixer(<left>)(<right>)(<bottom>){<label>}
elecmixersize=<num>
```

Hybrid components

```
\optfilter[<opt>](<in>)(<out>){<label>}
filtersize=<num>
filtertype=bandpass, bandstop, lowpass,
highpass
filterangle=<num>
FilterStyle <psstyle>
```

```
\fibercollimator(<in>)(<A>)(<B>)(<out>){<label>}
fibercolsize=<num> or <width> <height>
```

```
\optdetector[<opt>](<in>)(<out>){<label>}
detsize=<num> or <width> <height>
dettype=round, diode
DetectorStyle <psstyle>
```

Special nodes

```
\oenode{<node>}{<comp>}
namingscheme=old, new
showoptdots=true, false
compname=<string>
```

```
\oenodeRefA{<comp>}
\oenodeRefB{<comp>}
\oenodeTrefA{<comp>}
\oenodeTrefB{<comp>}
\oenodeCenter{<comp>}
\oenodeLabel{<comp>}
\oenodeExt{<comp>}
extnode=<refpoint>
extnodealign=rel, relative, abs, absolute
extnodes=<list>
```

```
\oenodeIfc{<num>}{<comp>}
\oenodeIn{<comp>}
\oenodeOut{<comp>}
```

```

\oenodeRotref{<comp>}
\oenodeBeam{<num>}
\oenodeBeamUp{<num>}
\oenodeBeamLow{<num>}
\oeBeamCenter{<num>}
\oeBeamVec{<num>}
\oeBeamVecUp{<num>}
\oeBeamVecLow{<num>}
\oeBeamVecMedian{<num>}

```

Connecting components

```

\drawbeam[<options>]{<obj1>}{<obj2>}...

    raytrace=true, false
    useNA=true, false
    n=<code>
    beampos=[<x> ]<y>
    beamangle=<pscode>
    beamalign=rel, relative, abs, absolute,
        firstcomp
    beampathskip=<num>
    beampathcount=<num>
    beaminside=true, false
    beaminsidefirst=true, false
    beaminsidelast=true, false
    allowbeaminside=true, false
    forcebeaminside=true, false
    startinsidecount=<num>
    stopinsidecount=<num>
    beammode=refl, trans, reflective, transmittive,
        auto
    beamnodealign=vec, conn, vector, connection

\optplane(<center>)
    beam=true, false
    Beam <psstyle>
    addtoBeam=<list>
    newBeam=<list>
    ArrowInsideMinLength=<pscode>
    ArrowInsideMaxLength=<pscode>
    fade <linestyle>
    fadeto=white, black, transparency

```

```

fadepoints=<num>
fadefuncname=gauss, linear, squared, exp,
    custom
fadefunc=<PS code>

```

```

\drawwidebeam[<options>]{<obj1>}{<obj2>}...
    beamwidth=<pscode>
    beamdiv=<pscode>
    pswarning=true, false
    savebeampoints=true, false, <int>
    loadbeampoints=true, false, <int>
    savebeam=true, false, <int>
    loadbeam=true, false, <int>
    startinside=true, false
    stopinside=true, false

```

```

\drawfiber[<options>]{<obj1>}{<obj2>}...
    fiberalign=rel, relative, center, abs,
        absolute
    fiberangleA=<num>
    fiberangleB=<num>
    startnode=auto, N, 1, 2, ...
    stopnode=auto, N, 1, 2, ...
    Fiber <psstyle>
    addtoFiber=<list>
    newFiber=<list>
    fiberstyle=<string>

```

```

\drawwire[<options>]{<obj1>}{<obj2>}...
    wirealign=rel, relative, center, abs,
        absolute
    wireangleA=<num>
    wireangleB=<num>
    wirestyle=<string>
    addtoWire=<list>
    newWire=<list>
    Wire <psstyle>
    fiber=[*+]none, all, i, o, <refpoint>
    wire=[*+]none, all, i, o, <refpoint>

```

```

\begin{optexp}...\end{optexp}
\backlayer{<code>}
\frontlayer{<code>}

```

Custom components

```

\optdipole[<options>](<in>)(<out>){<comp>}{<label>}
\opttripole[<options>](<in>)(<center>)(<out>){<comp>}{<la

    optdipolesize=<width>[ <height>]
    optdipolecomp=<macros>
    opttripolecomp=<macros>

```

```

\newOptexpDipole[<fixopt>]{<name>}{<dftopt>}
\newOptexpTripole[<fixopt>]{<name>}{<dftopt>}
\newOptexpFiberDipole[<fixopt>]{<name>}{<dftopt>}
\newOptexpElecDipole[<fixopt>]{<name>}{<dftopt>}

```

Additional information

```

showifcnodes=true, false
IfcNodeStyle <psstyle>
showinterfaces=true, false
IfcStyle <psstyle>

```