

singleline
Single-line diagrams with TikZ
v. 1.0.0

Author: Isabelle Brandherm

ibrandherm@ikp.tu-darmstadt.de

July 17, 2020

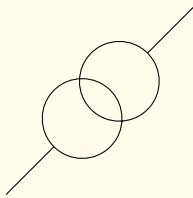
Contents

1	Introduction	2
2	Commands	2
2.1	Monopoles	2
2.2	Bipoles	4
2.2.1	Transformer	5
2.2.2	More bipoles	7
2.3	Shapes	9
2.4	Bus bar	10
3	Options	11
3.1	General options	11
3.2	Element size and scaling	13
4	Examples	14
4.1	9 Bus system	14
4.2	Shape orientation	15
5	Advanced settings	15
5.1	Defining global options	15
5.2	Pgfkey tree	16
5.3	Defining a new drawing command	18
5.4	Package requirements	19

1 Introduction

The **singleline** package is a **TikZ** based **L^AT_EX 2_ε** package for drawing single-line diagrams of three-phase power systems. It consists of a collection of drawing commands and nodes of the most used circuit symbols. The package is based on the EN 60617-2/-11 standard.

To use `singleline`-commands enclosure in a `tikzpicture` environment or usage as `\tikz` inline graphic is required. `singleline` should be fully compatible with `TikZ`, however conflicts with other circuit drawing packages, such as `CircuitikZ` cannot be excluded.



```
\begin{tikzpicture}
\trafo (0,0)(2.5,2.5);
\end{tikzpicture}
```

2 Commands

The **singleline** package consists mainly of a collection of drawing commands for circuit symbols. Most drawing commands can be classified in two categories: First so called monopoles and bipoles, explained in detail in the following sections. Second some additional circuit symbols and bus bar drawing are provided.

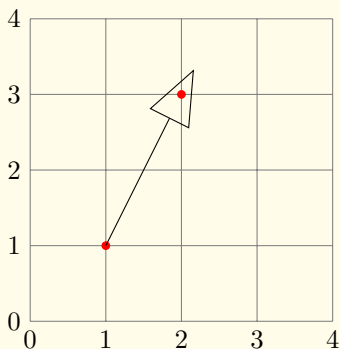
2.1 Monopoles

Monopoles are drawing commands where the circuit symbol is placed at the end of a line. The commands can be represented by the basic syntax pattern

$$\backslash\langle\text{command}\rangle[\langle\text{optional arguments}\rangle](\langle\text{coordinate1}\rangle)(\langle\text{coordinate2}\rangle);$$

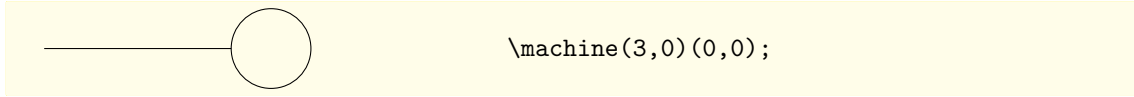
where coordinates are enclosed in parantheses, while optional arguments come as usual in square brackets. `TikZ` inherent actions on coordinates, such as naming or calculations are supported. For monopole commands, the circuit symbol, e.g. a load, is drawn as a `TikZ` node, centred on `<coordinate2>` whilst a straight line is drawn from `<coordinate1>` to the edge of the node. Thereby the node is rotated to align it to the vector from `<coordinate1>` to `<coordinate2>`.

Example:

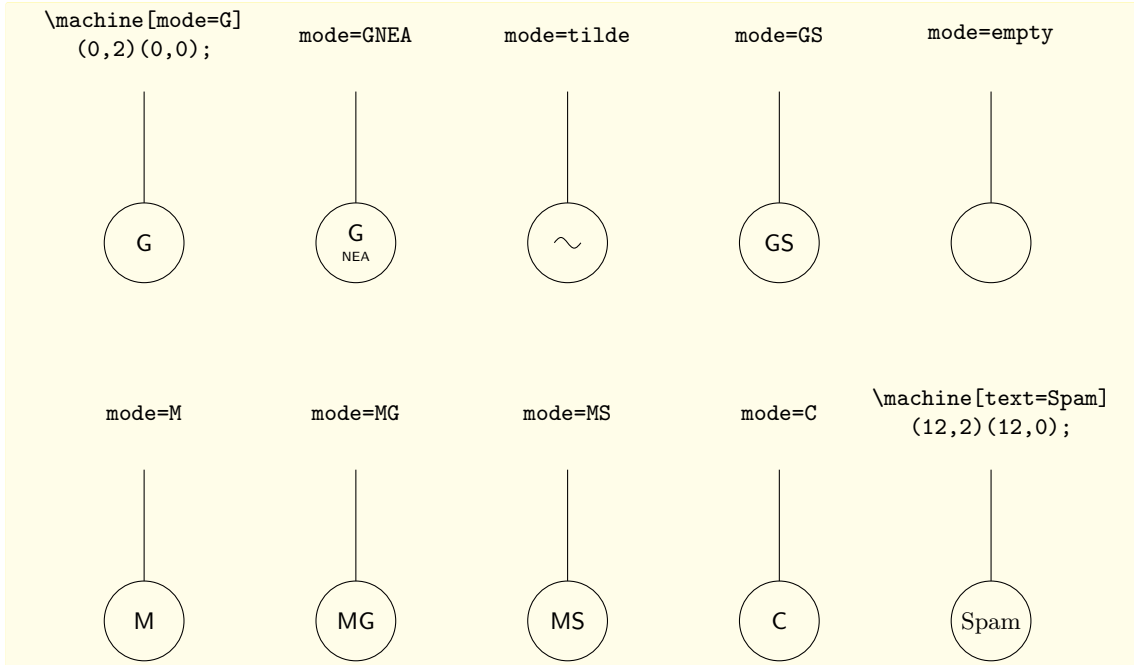


```
\begin{tikzpicture}
\draw[color=gray, thin] (0,0) grid (4,4);
\foreach \x in {0,...,4}{
\node[anchor=east] at (0,\x){\x};
\node[anchor=north] at (\x,0){\x};}
\draw[color=red, fill] (1,1) circle (0.05);
\draw[color=red, fill] (2,3) circle (0.05);
\elldoad(1,1)(2,3);
\end{tikzpicture}
```

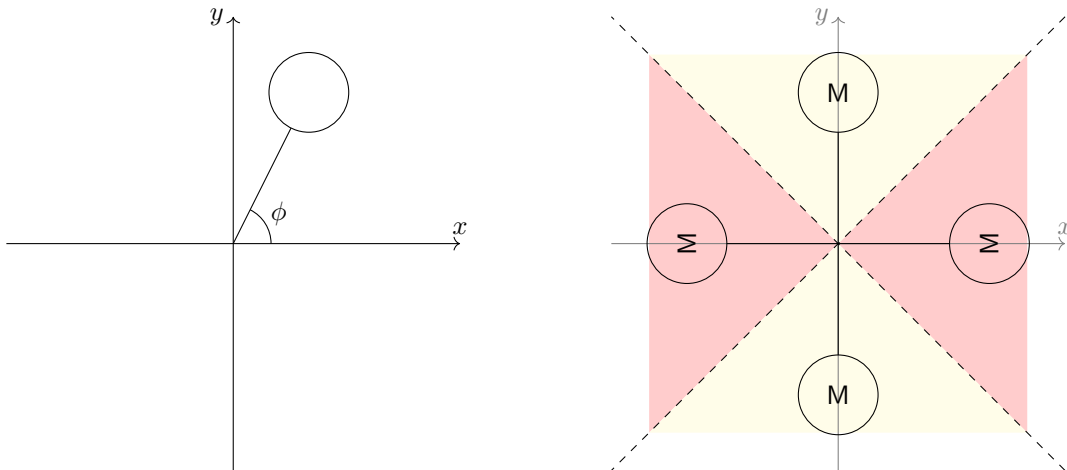
Machine `\machine(coordinate1)(coordinate2);`



The machine symbol is based on a simple circular shape with a centred node text. The node text can be accessed either by the option `mode` to use one of the predefined entries or by the option `text` to set the entry manually.

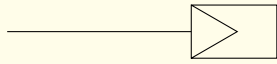


The rotational behaviour of the machines is somewhat different from other commands, to ensure a defined orientation of the node text.



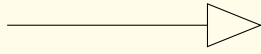
In the above figure, ϕ is the angle between the positive x -axis and the machine. The regions for the different text orientations are marked on the right hand side. In the yellow regions, where $45^\circ < \phi \leq 135^\circ$ and $225^\circ < \phi \leq 315^\circ$, the node content is equally oriented to the text direction. In the red marked regions, $0^\circ < \phi \leq 45^\circ$, $135^\circ < \phi \leq 225^\circ$ and $315^\circ < \phi \leq 360^\circ$ the node text is rotated counter clockwise by 90° .

Photovoltaic `\pv(coordinate1)(coordinate2);`

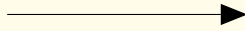


```
\pv(0,0)(3,0);
```

Load `\elload(coordinate1)(coordinate2);`

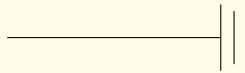


```
\elload(0,0)(3,0);
```



```
\elload[mode = small](0,0)(3,0);
```

Ground `\ground(coordinate1)(coordinate2);`



```
\ground(0,0)(3,0);
```

Grid `\elgrid(coordinate1)(coordinate2);`



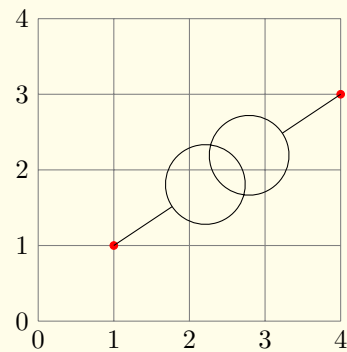
```
\elgrid(0,0)(3,0);
```

2.2 Bipoles

Circuit symbols at the center of a line are called bipoles. As monopoles they rely on the basic pattern

```
\<command>[<optional arguments>](<coordinate1>)(<coordinate2>);
```

. Here, the node is placed in the center between `coordinate1` and `coordinate2` and rotated accordingly.

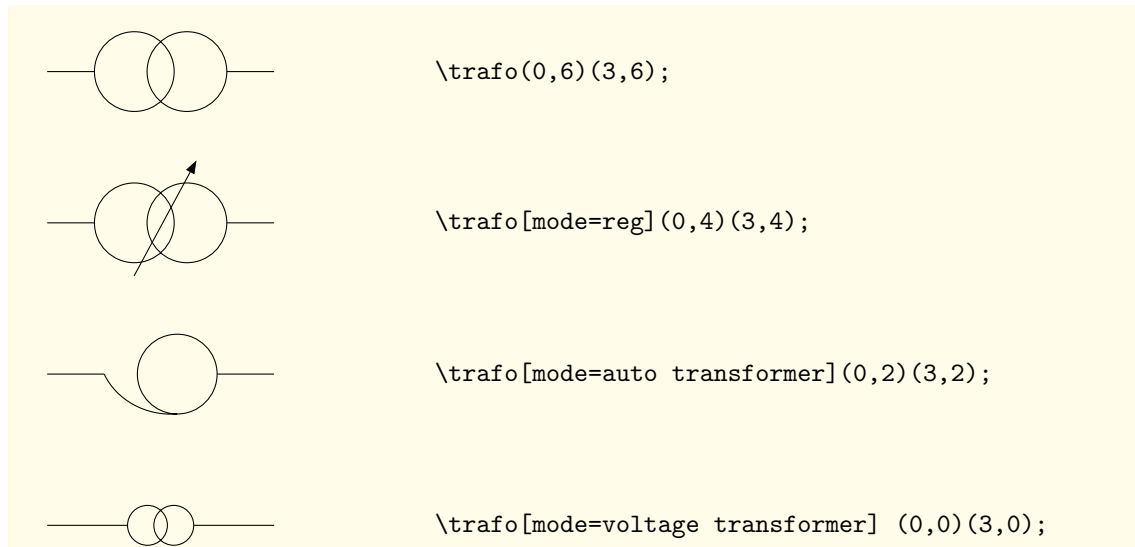


```
\begin{tikzpicture}
\draw[gray, thin] (0,0) grid (4,4);
\foreach \x in {0,...,4}{
\node[anchor=east] at (0,\x){\x};
\node[anchor=north] at (\x,0){\x};}
\draw[red, fill] (1,1) circle (0.05);
\draw[red, fill] (4,3) circle (0.05);
\trafo(1,1)(4,3);
\end{tikzpicture}
```

2.2.1 Transformer

```
\trafo(coordinate1)(coordinate1);
```




The transformer command is treated separately since it is provided with some additional options: three phase transformer connection, neutral point earthing and two color mode. First of all, there are some additional transformer variations besides the default shape:



The following options are created to the simple default transformer shape and might look weird in combination with different **modes** (but feel free to try).

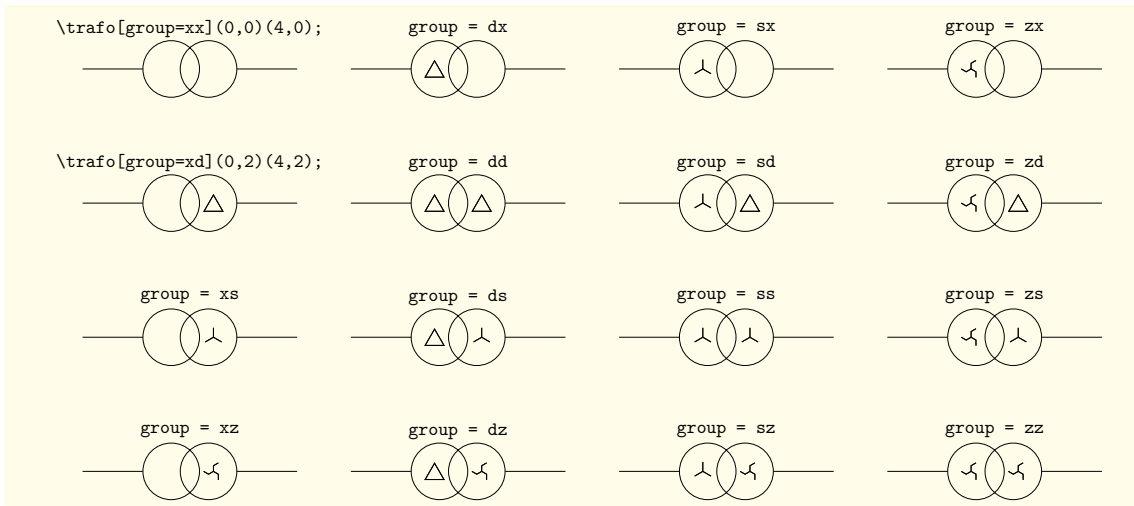
Three phase transformer connection Symbols for the three standard three phase transformer connections are defined as node shapes:

```
\node[draw, sltrafotriangle] at (0,0){};
```

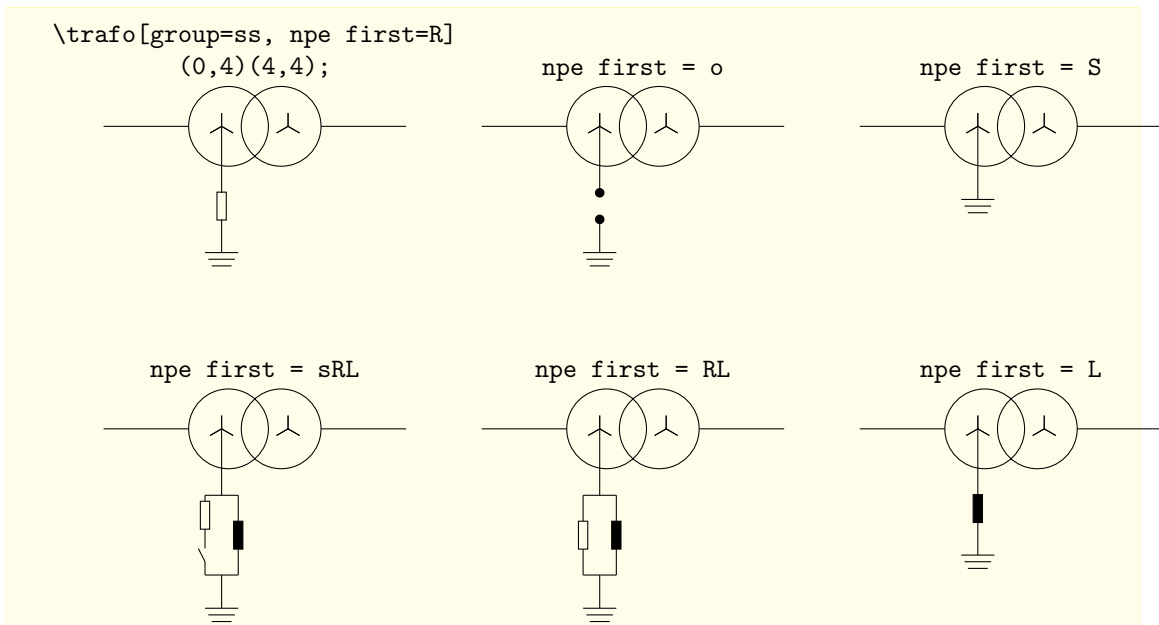
	<code>sltrafostar</code>	<code>sltrafozigzag</code>
		

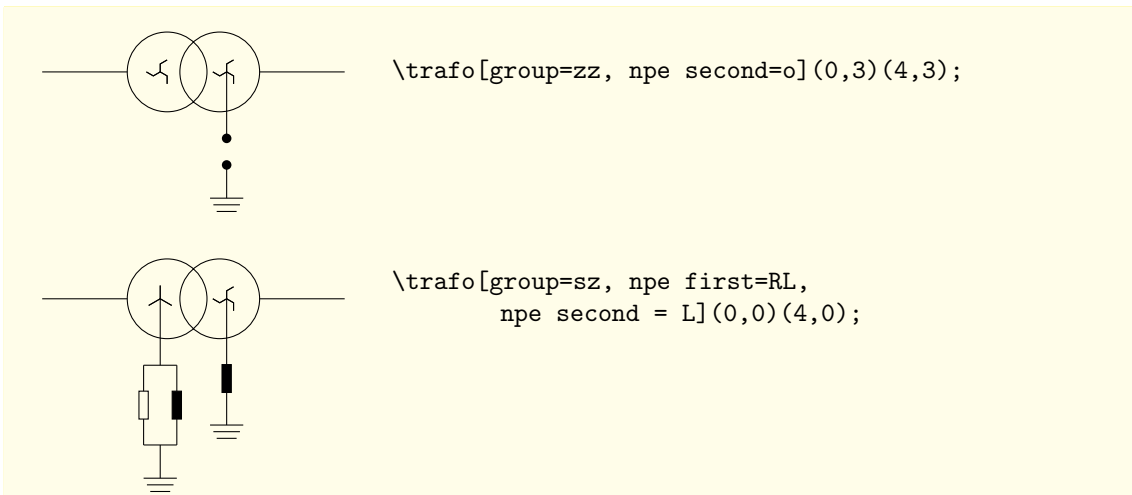
They can either be used as shown above as nodes or used within the transformer command with the keyword **group**, followed by a two letters. The first indicates the side of the trafo, which is next to the first coordinate, the letter represents the side next to the second coordinate. The letters are coded as follows:

x = no symbol	d = Δ - connection
s = star - connection	z = zigzag - connection

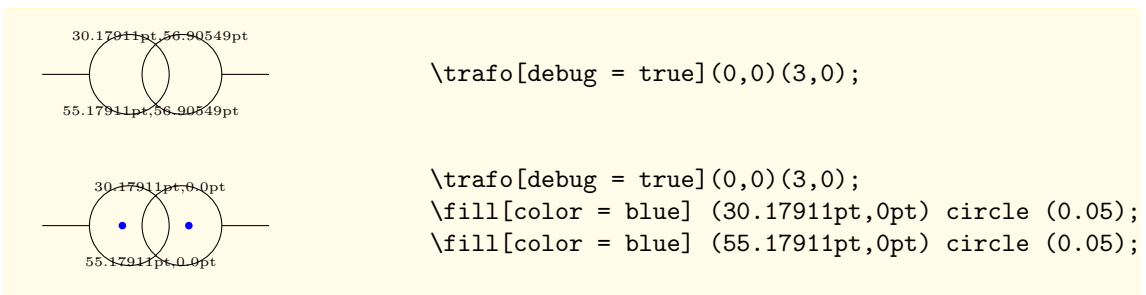


Neutral point earthing Six variations of neutral point earthing drawings are available. They can be drawn at the transformer side next to the first coordinate using the key `npe first`. The neutral point earthing of the second transformer half can be drawn with the option `npe second`. Both keys can be handled equally - here demonstrated for `npe first`.

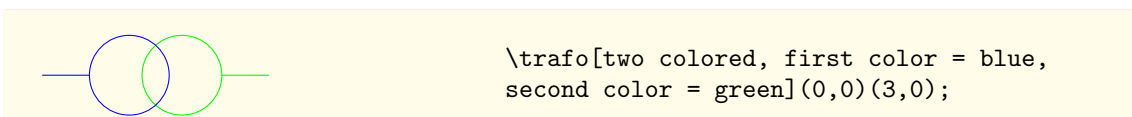




Debug mode The `debug` option of the transformer command prints the coordinates of the point in the transformer circles, where the transformer connections or neutral point earthing are anchored. It can e.g. be used for further neutral point and transformer connection customization.

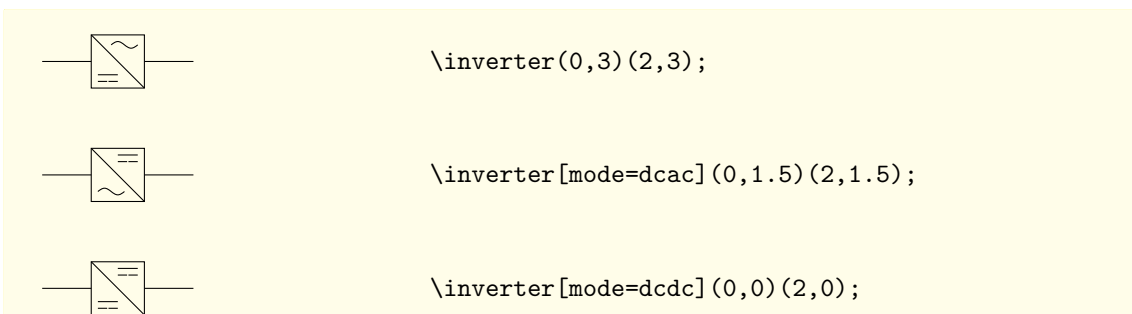


Two colored transformer In the current version, the two colored transformer option is only defined for the default transformer shape and cannot be combined with neutral point earthing or transformer connections. It can be switched on with the option `two colored`. The colors can then be passed to the command with the keys `first color` and `second color`. If the colors aren't defined, the current active color is used.

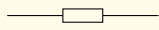


2.2.2 More bipoles

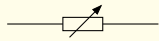
Inverter `\inverter(coordinate1)(coordinate2);`



Resistor `\resistor(coordinate1)(coordinate2);`



`\resistor(0,1.5)(2,1.5)`

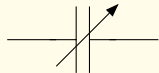


`\resistor[mode=reg](0,0)(2,0);`

Capacitor `\capacitor(coordinate1)(coordinate2);`

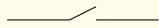


`\capacitor(0,1.5)(2,1.5);`

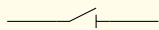


`\capacitor[mode=reg](0,0)(2,0);`

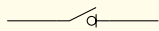
Switch `\switch(coordinate1)(coordinate2);`



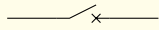
`\switch(0,4.5)(2,4.5);`



`\switch[mode = disconnect](0,3)(2,3);`

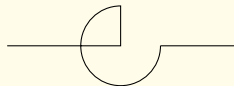


`\switch[mode = load-break](0,1.5)(2,1.5);`

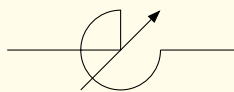


`\switch[mode = circuit breaker](0,0)(2,0);`

Reactor `\reactor(coordinate1)(coordinate2);`



`\reactor(0,2)(3,2);`

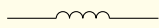


`\reactor[mode=reg](0,0)(3,0);`

Inductance `\reactor(coordinate1)(coordinate2);`

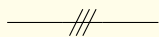


`\inductance(0,1.5)(2,1.5);`



`\inductance[mode = coil](0,0)(2,0);`


Three phase line `\linetp(coordinate1)(coordinate2);`




`\linetp(0,0)(2,0);`

2.3 Shapes


Besides the shapes used in the commands, the following shapes are defined:



```
\node[draw,fault] at (0,4){};
```

```
\node[draw,currenttransformer] at (0,2){};
```

```
\node[draw,currenttransformer] at (0,0.5){};
\draw(0,1)--(0,0);
```

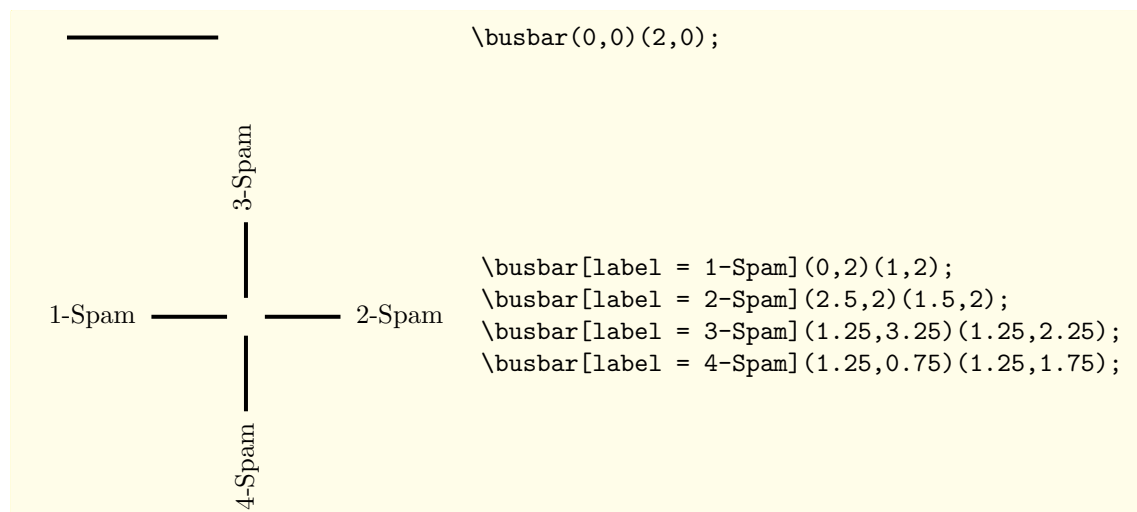
In the table below, the identifiers for shapes used for drawing commands are listed. They can also separately be drawn as nodes. The identifiers are prefixed with "sl" (for single-line), to avoid internal errors.

Command	Mode	Shape identifier
\machine	all modes	slcircle
\pv		slpv
\load		slload
\ground		slground
\elgrid		slgrid
\trafo	default	sltrafo
	reg	slregtrafo
	autransformer	slautotransformer
	two colored transformer	sltwocoloredtrafo
	voltage transformer	slvoltagegetransformer
\inverter	default	sldcac
	acdc	slacdc
	dcdc	sldcdc
\resistor	default	slresistor
	reg	slregresistor
\capacitor	default	slcapacitor
	reg	slregcapacitor
\switch	default	slswitch
	circuit breaker	slcircbreaker
	load-break	slloadbreakswitch
	disconnect	sldisconswitch
\reactor	default	slreactor
	reg	slregreactor
\inductance	default	slimpedance
	coil	slinductance
\linetp	default	sllinetp

2.4 Bus bar

```
\busbar(coordinate 1)(coordinate 2);
```

The default line width of bus bars is $0.3*\texttt{\elsize}$. The bus bar `label` option defines label text position and orientation.



3 Options

3.1 General options

Connectors Connectors are basically arrow tips, which are used to connect lines and circuit commands. As a consequence, they can be interchanged with other *TikZ* based arrows. The connectors are named with a three-letter scheme:

First letter	Second letter	Third letter
c = closed	c = closed	t = arrow points to the connector
o = open	o = open	f = arrow points away from the connector
x = not existent	x = not existent	x = not existent

cxx-		cxt-		cx-	
ccx-		cct-		ccf-	
cox-		cot-		cof-	
oxx-		oxt-		oxf-	
ocx-		oct-		ocf-	
oox-		oot-		oof-	
xcx-		xct-		xcf-	
xox-		xot-		xof-	
		xxt-		xxf-	

	<code>\inverter[connector=ccx](0,4)(2,4);</code>
	<code>\inverter[connector=ccx](0,2)(2,2);</code>
	<code>\inverter[connector=ccx](0,0)(2,0);</code>
	<code>\draw[ccx-ccx] (0,0)--(2,0);</code>

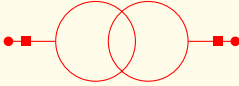
Color



```
\machine[color=green!50!black](0,2)(3,2);
```

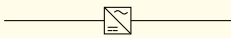


```
\inverter[color=blue](0,0)(3,0);
```



```
\trafo[connector=ccx-ccx, color=red](0,0)(3,0);
```

Scale `scale` is only designed for shape scaling. Line width and connectors can be scaled using the `line width` key.



```
\inverter[scale=0.5](0,4)(3,4);
```

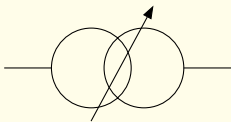


```
\inverter(0,2)(3,2);
```

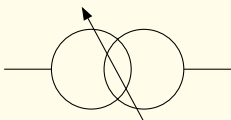


```
\inverter[scale=1.5](0,0)(3,0);
```

Mirror The shape of the element is mirrored along the axis perpendicular to the drawing axis.



```
\trafo[mode=reg](0,2)(3,2);
```



```
\trafo[mode=reg, mirror=true](0,0)(3,0);
```

Line width



```
\inverter(0,2)(3,2);
```

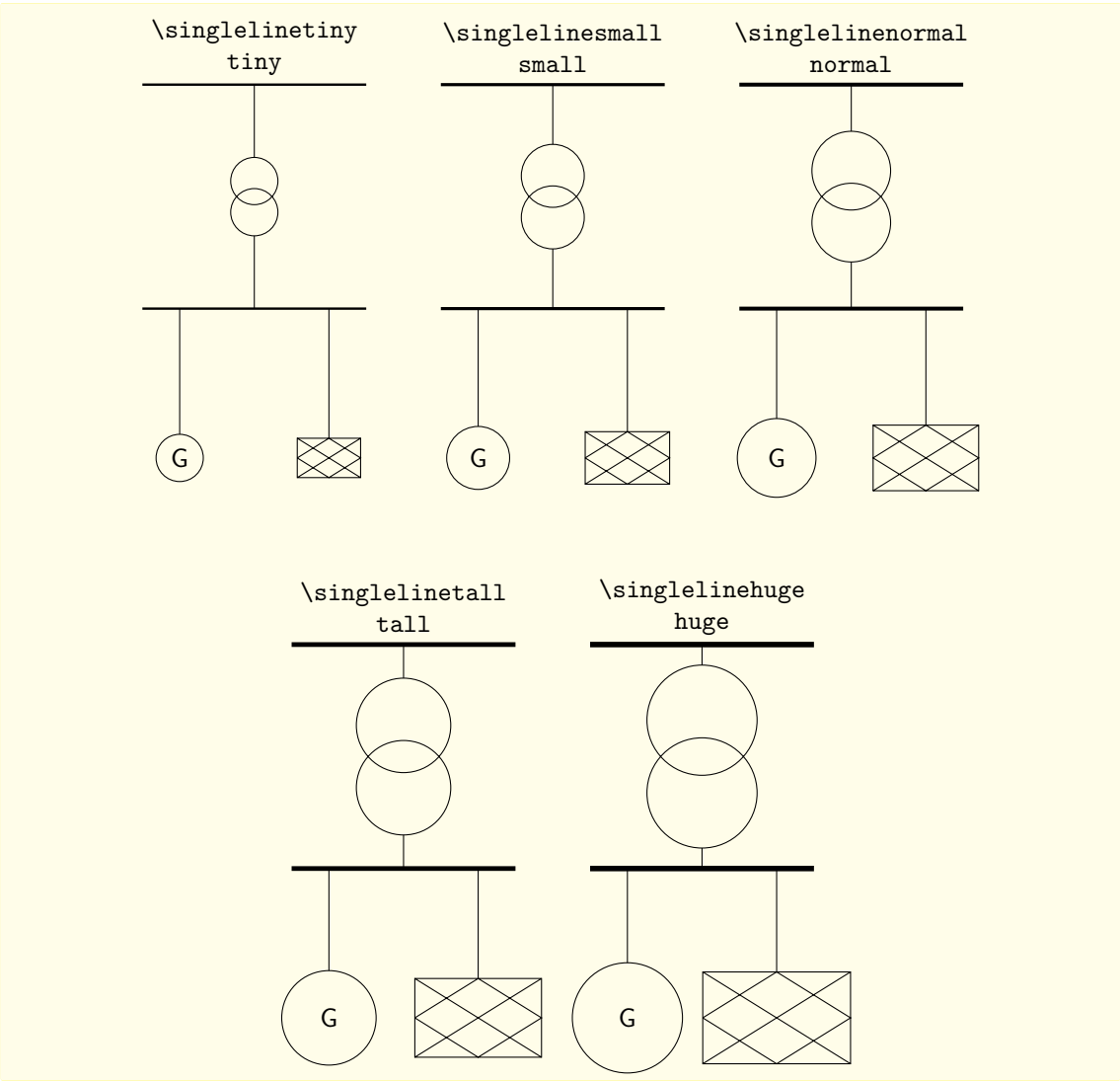


```
\inverter[line width=1.5pt](0,0)(3,0);
```

3.2 Element size and scaling

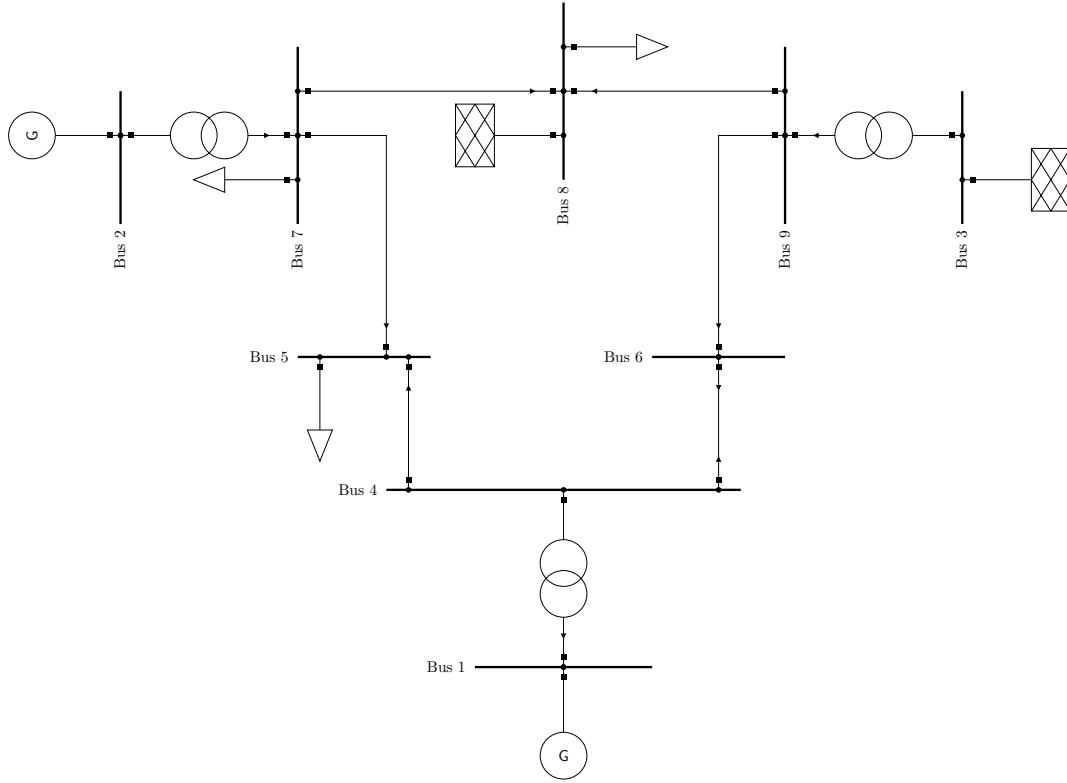
Every shapes is defined using a reference size, called `\elsize`. Changing the `\elsize` thus changes the size of all shapes defined in the `singleline` package and should be used carefully. However, there are some predefined settings for the `elsize`, which can be called globally as package options, e.g. `\usepackage[small]{singleline}` or local with the commands in the table below.

Command	Package option	<code>elsize</code>
<code>\singlelinetiny</code>	<code>tiny</code>	3pt
<code>\singlelinesmall</code>	<code>small</code>	4pt
<code>\singlelinenormal</code>	<code>normal</code> (default)	5pt
<code>\singlelinetall</code>	<code>tall</code>	6pt
<code>\singlelinehuge</code>	<code>huge</code>	7pt



4 Examples

4.1 9 Bus system



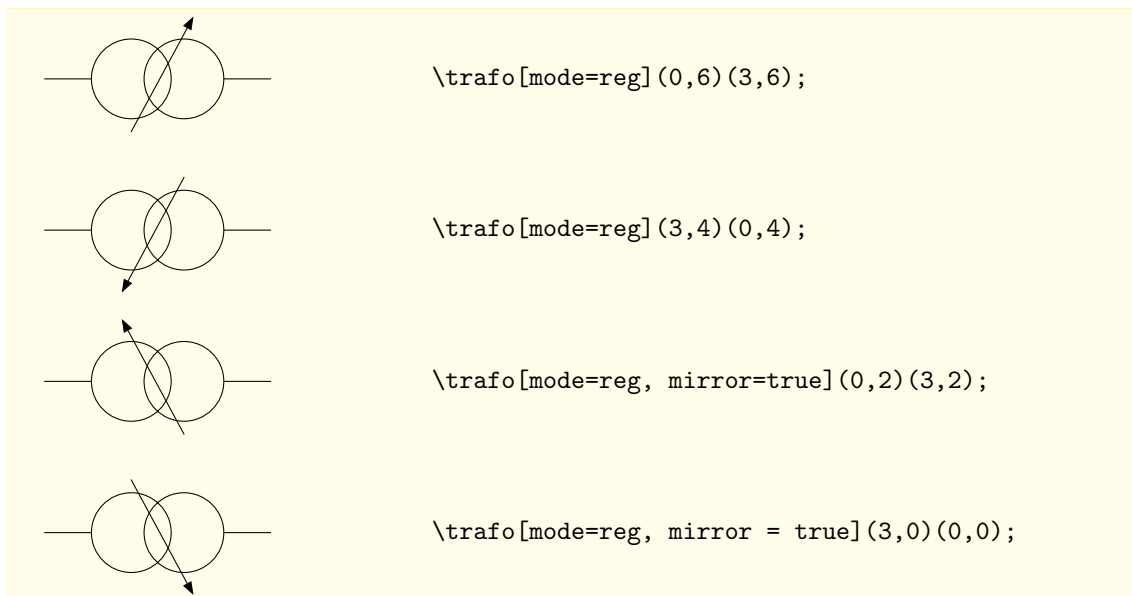
```

\begin{tikzpicture}
%
\busbar[label=Bus 1](8,-1)(12,-1);
\busbar[label=Bus 2](0,9)(0,12);
\busbar[label=Bus 3](19,9)(19,12);
\busbar[label=Bus 4](6,3)(14,3);
\busbar[label=Bus 5](4,6)(7,6);
\busbar[label=Bus 6](12,6)(15,6);
\busbar[label=Bus 7](4,9)(4,13);
\busbar[label=Bus 8](10,10)(10,14);
\busbar[label=Bus 9](15,9)(15,13);
%
\machine[connector=ccx-,mode=G](10,-1)($(10,-1)-(0,2)$);
\machine[connector=ccx-, mode=G](0,11)($(0,11)-(2,0)$);
\elload[connector=ccx-](4.5,6)($(4.5,6)-(0,2)$);
\elload[connector=ccx-](4,10)($(4,10)-(2,0)$);
\elload[connector=ccx-](10,13)($(10,13)+(2,0)$);
\elgrid[connector=ccx-](19,10)($(19,10)+(2,0)$);
\elgrid[connector=ccx-](10,11)($(10,11)-(2,0)$);
\trafo[connector=cct-ccx](10,-1)(10,3);
\trafo[connector=cct-ccx](4,11)(0,11);
\trafo[connector=cct-ccx](15,11)(19,11);
\draw[cct-ccx](6.5,6)--(6.5,3);
\draw[ccf-ccf](13.5,6)--(13.5,3);
\draw[ccx-cct](4,11)-|(6,6);
\draw[ccx-cct](15,11)-|(13.5,6);
\draw[ccx-cct](4,12)--(10,12);
\draw[cct-ccx](10,12)--(15,12);
%
\end{tikzpicture}

```

4.2 Shape orientation

Sometimes, the default orientation of a circuit symbol in a command might not be ideal. Symbols can simply be rotated by 180 degrees by interchanging the first and second coordinate of the command. In combination with the previously mentioned mirror option, the desired orientation should be possible.

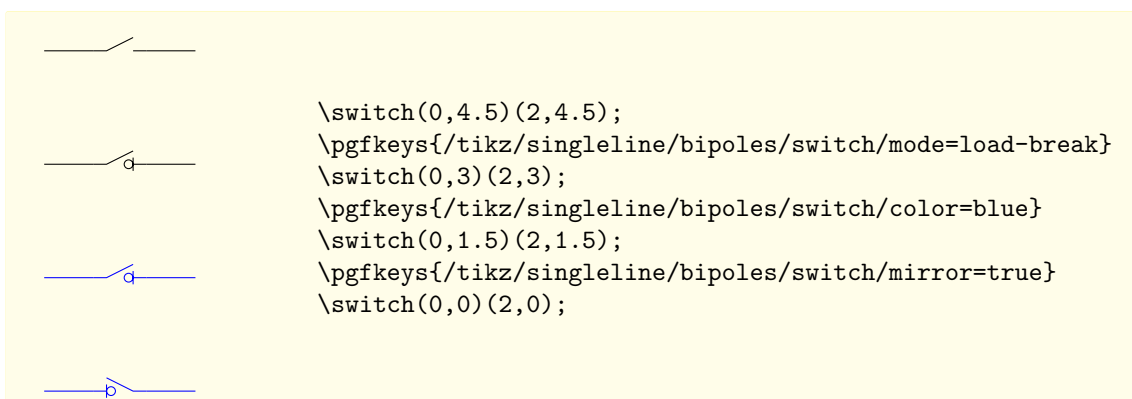


5 Advanced settings

5.1 Defining global options

The `singeline` package uses, as `TikZ`, heavily the `pgfkey` mechanism. The default values for commands and their optional arguments are stored in a `pgfkey` tree. By altering the key values, options can be set globally for the entire document, or if used in some sort of scoping environment, for a section. Detailed information about the `pgf` keys can be found in the `pgf` documentation. For `singelinetikz` defined keys can be found in the next section. The values of the keys can be manipulated in the easiest way with `\pgfkeys{}`.

Example



5.2 Pgfkey tree

Machine

```
/tikz/singleline/monopoles/machine
/tikz/singleline/monopoles/machine/color
/tikz/singleline/monopoles/machine/scale
/tikz/singleline/monopoles/machine/line width
/tikz/singleline/monopoles/machine/label
/tikz/singleline/monopoles/machine/mode
/tikz/singleline/monopoles/machine/text
/tikz/singleline/monopoles/machine/shape
/tikz/singleline/monopoles/machine/connector
/tikz/singleline/monopoles/machine/leftconnector
/tikz/singleline/monopoles/machine/rightconnector
```

Photovoltaic

```
/tikz/singleline/monopoles/pv
/tikz/singleline/monopoles/pv/color
/tikz/singleline/monopoles/pv/scale
/tikz/singleline/monopoles/pv/line width
/tikz/singleline/monopoles/pv/label
/tikz/singleline/monopoles/pv/shape
/tikz/singleline/monopoles/pv/connector
/tikz/singleline/monopoles/pv/leftconnector
/tikz/singleline/monopoles/pv/rightconnector
```

Load

```
/tikz/singleline/monopoles/load
/tikz/singleline/monopoles/load/color
/tikz/singleline/monopoles/load/scale
/tikz/singleline/monopoles/load/line width
/tikz/singleline/monopoles/load/label
/tikz/singleline/monopoles/load/mode
/tikz/singleline/monopoles/load/shape
/tikz/singleline/monopoles/load/connector
/tikz/singleline/monopoles/load/leftconnector
/tikz/singleline/monopoles/load/rightconnector
```

Ground

```
/tikz/singleline/monopoles/ground
/tikz/singleline/monopoles/ground/color
/tikz/singleline/monopoles/ground/scale
/tikz/singleline/monopoles/ground/line width
/tikz/singleline/monopoles/ground/label
/tikz/singleline/monopoles/ground/shape
/tikz/singleline/monopoles/ground/connector
/tikz/singleline/monopoles/ground/leftconnector
/tikz/singleline/monopoles/ground/rightconnector
```

Grid

```
/tikz/singleline/monopoles/grid
/tikz/singleline/monopoles/grid/color
/tikz/singleline/monopoles/grid/scale
/tikz/singleline/monopoles/grid/line width
/tikz/singleline/monopoles/grid/label
/tikz/singleline/monopoles/grid/shape
/tikz/singleline/monopoles/grid/connector
/tikz/singleline/monopoles/grid/leftconnector
/tikz/singleline/monopoles/grid/rightconnector
```

Transformer

```
/tikz/singleline/bipoles/trafo
/tikz/singleline/bipoles/trafo/color
/tikz/singleline/bipoles/trafo/scale
/tikz/singleline/bipoles/trafo/line width
```

```
/tikz/singleline/bipoles/trafo/label
/tikz/singleline/bipoles/trafo/shape
/tikz/singleline/bipoles/trafo/connector
/tikz/singleline/bipoles/trafo/leftconnector
/tikz/singleline/bipoles/trafo/rightconnector
/tikz/singleline/bipoles/trafo/group
/tikz/singleline/bipoles/trafo/group draw
/tikz/singleline/bipoles/trafo/debug
/tikz/singleline/bipoles/trafo/npe first
/tikz/singleline/bipoles/trafo/npe draw first
/tikz/singleline/bipoles/trafo/npe second
/tikz/singleline/bipoles/trafo/npe draw second
/tikz/singleline/bipoles/trafo/two colored
/tikz/singleline/bipoles/trafo/first color
/tikz/singleline/bipoles/trafo/second color
```

Capacitor

```
/tikz/singleline/bipoles/capacitor
/tikz/singleline/bipoles/capacitor/color
/tikz/singleline/bipoles/capacitor/label
/tikz/singleline/bipoles/capacitor/line width
/tikz/singleline/bipoles/capacitor/connector
/tikz/singleline/bipoles/capacitor/leftconnector
/tikz/singleline/bipoles/capacitor/rightconnector
/tikz/singleline/bipoles/capacitor/shape
/tikz/singleline/bipoles/capacitor/scale
/tikz/singleline/bipoles/capacitor/mirror
/tikz/singleline/bipoles/capacitor/mode
```

Inverter

```
/tikz/singleline/bipoles/inverter
/tikz/singleline/bipoles/inverter/color
/tikz/singleline/bipoles/inverter/label
/tikz/singleline/bipoles/inverter/line width
/tikz/singleline/bipoles/inverter/connector
/tikz/singleline/bipoles/inverter/leftconnector
/tikz/singleline/bipoles/inverter/rightconnector
/tikz/singleline/bipoles/inverter/shape
/tikz/singleline/bipoles/inverter/scale
/tikz/singleline/bipoles/inverter/mirror
/tikz/singleline/bipoles/inverter/mode
```

Inductance

```
/tikz/singleline/bipoles/inductance
/tikz/singleline/bipoles/inductance/color
/tikz/singleline/bipoles/inductance/label
/tikz/singleline/bipoles/inductance/line width
/tikz/singleline/bipoles/inductance/connector
/tikz/singleline/bipoles/inductance/leftconnector
/tikz/singleline/bipoles/inductance/rightconnector
/tikz/singleline/bipoles/inductance/shape
/tikz/singleline/bipoles/inductance/scale
/tikz/singleline/bipoles/inductance/mirror
/tikz/singleline/bipoles/inductance/mode
```

Three phase line

```
/tikz/singleline/bipoles/linetp
/tikz/singleline/bipoles/linetp/color
/tikz/singleline/bipoles/linetp/label
/tikz/singleline/bipoles/linetp/line width
/tikz/singleline/bipoles/linetp/connector
/tikz/singleline/bipoles/linetp/leftconnector
/tikz/singleline/bipoles/linetp/rightconnector
/tikz/singleline/bipoles/linetp/shape
/tikz/singleline/bipoles/linetp/scale
/tikz/singleline/bipoles/linetp/mirror
```


Resistor

```
/tikz/singleline/bipoles/resistor
/tikz/singleline/bipoles/resistor/color
/tikz/singleline/bipoles/resistor/label
/tikz/singleline/bipoles/resistor/line width
/tikz/singleline/bipoles/resistor/connector
/tikz/singleline/bipoles/resistor/leftconnector
/tikz/singleline/bipoles/resistor/rightconnector
/tikz/singleline/bipoles/resistor/shape
/tikz/singleline/bipoles/resistor/scale
/tikz/singleline/bipoles/resistor/mirror
/tikz/singleline/bipoles/resistor/mode
```

Switch

```
/tikz/singleline/bipoles/switch
/tikz/singleline/bipoles/switch/color
/tikz/singleline/bipoles/switch/label
/tikz/singleline/bipoles/switch/line width
/tikz/singleline/bipoles/switch/connector
/tikz/singleline/bipoles/switch/leftconnector
/tikz/singleline/bipoles/switch/rightconnector
/tikz/singleline/bipoles/switch/shape
/tikz/singleline/bipoles/switch/scale
/tikz/singleline/bipoles/switch/mirror
/tikz/singleline/bipoles/switch/mode
```

Reactor

```
/tikz/singleline/bipoles/reactor
/tikz/singleline/bipoles/reactor/color
/tikz/singleline/bipoles/reactor/label
/tikz/singleline/bipoles/reactor/line width
/tikz/singleline/bipoles/reactor/connector
/tikz/singleline/bipoles/reactor/leftconnector
/tikz/singleline/bipoles/reactor/rightconnector
/tikz/singleline/bipoles/reactor/shape
/tikz/singleline/bipoles/reactor/scale
/tikz/singleline/bipoles/reactor/mirror
/tikz/singleline/bipoles/reactor/mode
```

Bus bar

```
/tikz/singleline/busbar/busbar line width
/tikz/singleline/busbar/label
/tikz/singleline/busbar/color
/tikz/singleline/busbar/leftconnector
/tikz/singleline/busbar/rightconnector
/tikz/singleline/busbar/connector
```

- `.../color`: Defines drawing color. Default: `.`
- `.../scale`: Defines scale factor. Default: `1`
- `.../shape`: Stores the used shape. Default: see section 2.3.
- `.../line width`: Defines line width. Default: `0.4pt`.
- `.../label`: Sets label. Default: empty
- `...mode`: The `mode` keys are created as `.is choice-` keys. The choice sets the value of the `shape`-key to the corresponding node shape.
- `.../trafo/group`: Is an `.is choice`-key, that sets the value of `.../trafo/group draw` according to the selected sub-key.
- `.../trafo/group draw`: Contains the drawing commands for the transformer three phase connections (see section 2.2.1).
- `.../trafo/npe first`: Is an `.is choice` key, that sets the value of `.../trafo/npe draw first`
- `.../trafo/npe draw first` stores the drawing commands for the neutral pint earthing on the first transformer half.
- `.../trafo/npe second` and `.../trafo/npe draw second`: As `draw first`, but for the second transformer side.
- `.../trafo/two colored` Turns on the two colored mode.
- `.../trafo/first color` and `.../trafo/second color` Set colors for two colored transformer mode. Default: `.`
- `.../busbar/busbar line width`: Line width of bus bars. Default: `0.3*\elsize`.
- `.../busbar/label`: Bus bar label.

5.3 Defining a new drawing command

Defining new drawing commands with this package is quite simple. Most circuit symbol commands rely on two basic commands, one is used for the definition of monopoles and the other for bipoles. These commands basically support the "standard" drawing options that have been listed in section 3.1. Commands as `\trafo` with extended options are defined separately. The basic commands can be understood as an input mask, that reads the specified information from a previously defined pgf key tree. Assume, one wants to create a new bipole `\foo` with a simple round node at the center. The first step is to create the necessary pgf tree and define the default values. For a bipole the following form is required:

```
\pgfkeysdefargs{/tikz/singleline/bipoles/foo/connector}{#1-#2}{
    \pgfkeys{/tikz/singleline/bipoles/foo/leftconnector=#1-}
    \pgfkeys{/tikz/singleline/bipoles/foo/rightconnector=-#2}}
\pgfkeys{/tikz/singleline/bipoles/foo/.cd,
color/.initial={blue},
label/.initial={},
line width/.initial={2pt},
leftconnector/.initial={},
rightconnector/.initial={},
connector/.initial={ - },
shape/.initial={circle},
scale/.initial={1},
mirror/.initial={false},
mode/.is choice,}
```

Afterwards the command can be defined using the macro

```
\NewDocumentCommand{\foo}{0}{r()r()}{
\SinglelineDefineBipoleCommands[#1]{foo}{#2}{#3}} .
```

If one appends these lines in the `singleline.sty` document and reinstalls the package, one should be able to generate this output without errors:



```
\begin{tikzpicture}
\foo(0,0)(2,0);
\end{tikzpicture}
```

Defining monopoles works quite similar. The structure of the needed keys and the definition macro can be found in the `singleline.sty` document.

5.4 Package requirements

The following packages must be installed for the usage of the `singleline` package:

- tikz
- xparse
- ifthenelse
- pgf
- pgfmath

In addition, the following *TikZ* libraries are used:

- arrows.meta
- calc
- positioning