

The **string-diagrams** package*

Draw string diagrams using TikZ

Paolo Brasolin
paolo.brasolin@gmail.com

v0.2.0 (2023/06/12)

Please note this is the **major version zero**, meant for initial development: *anything MAY change at any time*. The upside is that this is the best time to **contribute**! Of course you can also just keep the **sty** along with your code and not care at all.

1 Documentation

/pgf/box

New: 2023-05-31
Updated: 2023-06-12

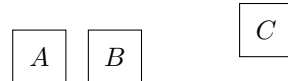
To draw boxes, you use this style on a node.

```
\begin{tikzpicture}  
  \node[box] {A};  
\end{tikzpicture}
```



You can draw multiple boxes using any of your standard TikZ positioning techniques. Don't forget to label the nodes so you can easily reference them.

```
\begin{tikzpicture}  
  \node[box] (A) at (0,0) {A};  
  \node[box, right of=A] (B) {B};  
  \node[box] (C) at ($(B)+(2cm,1em)$) {C};  
\end{tikzpicture}
```



*Thanks!

```

/pgf/box_ports_north /pgf/box ports north=<integer>
/pgf/box_ports_east  /pgf/box ports east=<integer>
/pgf/box_ports_south /pgf/box ports south=<integer>
/pgf/box_ports_west  /pgf/box ports west=<integer>

```

New: 2023-06-12

You can open up any number of ports on any side of a box using the appropriate key. Then, you can refer to the opened ports by their index.

```

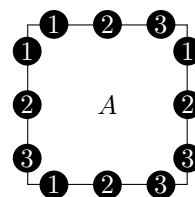
\begin{tikzpicture}[
  marker/.style={circle, fill, inner sep=1pt, text=white},
]

\node[
  box,
  box ports north=3,
  box ports east=3,
  box ports south=3,
  box ports west=3,
  minimum width=6em,
  minimum height=6em,
] (A) {A};

\foreach \side in {north,east,south,west}
  \foreach \index in {1,...,3}
    \node[marker] at (A.\side.\index) {\index};

\end{tikzpicture}

```



```

/pgf/box_ports /pgf/box ports=<integer>/<integer>/<integer>/<integer>

```

New: 2023-06-12

The `box ports` key is a shortcut to set the number of ports on all sides at once.

```

\begin{tikzpicture}[
  marker/.style={circle, fill, inner sep=1pt},
]

\node[box, box ports=1/2/3/4] (A) {A};

\foreach \side/\n in {north/1,east/2,south/3,west/4}
  \foreach \index in {1,...,\n}
    \node[marker] at (A.\side.\index) {};

\end{tikzpicture}

```

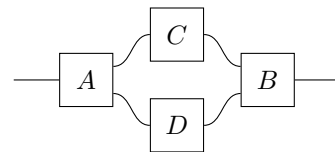


The same value can also be passed to the `box` key itself.

`\wires` `\wires[<TikZ keys>]{<connectivity>}{<loose ends>}`

New: 2023-05-31 To connect boxes, you can use the `\wires` macro. The first argument is `TikZ` styling for the wires; the second argument is a nested dictionary specifying the connectivity; the third argument is a list of the loose ends to draw. `boxes` have the following anchors: `west`, `west.0`, `west.1`, `east`, `east.0`, and `east.1`.

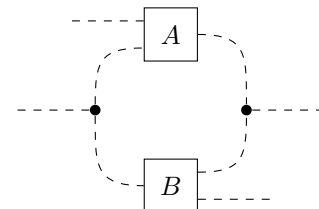
```
\begin{tikzpicture}[scale=0.6]
  \node[box=0/2/0/1] (A) at (-2, 0) {A};
  \node[box=0/1/0/2] (B) at (+2, 0) {B};
  \node[box=0/1/0/1] (C) at ( 0,+1) {C};
  \node[box=0/1/0/1] (D) at ( 0,-1) {D};
  \wires[] {
    A = { east.1 = C.west, east.2 = D.west },
    C = { east = B.west.1 },
    D = { east = B.west.2 },
  } { A.west, B.east }
\end{tikzpicture}
```



`/pgf/dot`

New: 2023-05-31 To split and join wires, you can use `dots` and their anchors `north`, `east`, `south`, and `west`. Remember to have fun with styling wires.

```
\begin{tikzpicture}
  \node[box=0/1/0/2] (A) at ( 0,+1) {A};
  \node[box=0/2/0/1] (B) at ( 0,-1) {B};
  \node[dot] (x) at (+1, 0) {};
  \node[dot] (y) at (-1, 0) {};
  \wires[looseness=1.5, dashed]{
    A = { east = x.north },
    B = { east.1 = x.south },
    y = { north = A.west.2, south = B.west },
  } {
    A.west.1, B.east.2, x.east, y.west
  }
\end{tikzpicture}
```



That's it. This is the package, for now.

2 Implementation

Open the `DocStrip` guards and set the internal namespace prefix (as per `LATEX3` `DocStrip` convention).

```
1 <*package>
2 <@@=stridi>
```

Load the essential support (`expl3`) “up-front”.

```
3 \RequirePackage{expl3}[2023/05/11]
4 \RequirePackage{tikz}[2023/01/15]
```

Identify the package and give the over all version information.

```

5 \ProvidesExplPackage
6   {string-diagrams}
7   {2023/06/12}
8   {0.2.0}
9   {Draw string diagrams using TikZ}

```

`/pgf/box_ports_north` Define high level keys to configure the number of ports on each side.

```

/pgf/box_ports_east
/pgf/box_ports_south
/pgf/box_ports_west
/pgf/box_ports
10 \pgfkeys{
11   /pgf/box-ports-north/.initial=1,
12   /pgf/box-ports-east/.initial=1,
13   /pgf/box-ports-south/.initial=1,
14   /pgf/box-ports-west/.initial=1,
15   /pgf/box-ports/.style-args={#1/#2/#3/#4}{
16     /pgf/box-ports-north=#1,
17     /pgf/box-ports-east=#2,
18     /pgf/box-ports-south=#3,
19     /pgf/box-ports-west=#4,
20   },
21 }

```

(End of definition for `/pgf/box ports north` and others. These functions are documented on page 2.)

`_stridi_intersect_hv_lines_through:NN` Calculates the intersection of two lines parallel to axes passing through given points on the plane.

#1 : Point through which the vertical line passes
 #2 : Point through which the horizontal line passes

```

22 \cs_new:Nn \_stridi_intersect_hv_lines_through:NN {
23   \pgfextractx { \pgf@xa } { #1 }
24   \pgfextracty { \pgf@ya } { #2 }
25   \pgfpoint { \pgf@xa } { \pgf@ya }
26 }

```

(End of definition for `_stridi_intersect_hv_lines_through:NN`.)

`_stridi_subdivide_segment:nNNNNN` Defines macros numbering equally spaced points on a segment.

#1 : Base namespace
 #2 : Points count
 #3 : Point containing the x coordinate of the starting point
 #4 : Point containing the y coordinate of the starting point
 #5 : Point containing the x coordinate of the ending point
 #6 : Point containing the y coordinate of the ending point

```

27 \cs_new:Nn \_stridi_subdivide_segment:nNNNNN {
28   \int_step_inline:nnnn { #2 } { -1 } { 1 } {
29     \cs_if_exist:cTF
30       { #1.#1 }
31       { \prg_break: }
32       { \prg_do_nothing: }

```

```

33 \cs_new_nopar:cpn
34 { #1.##1 }
35 {
36 \pgfmathdivide
37 { 2 * ##1 - 1 }
38 { 2 * #2 }
39 \pgfpointlineattime
40 { \pgfmathresult }
41 { \_stridi_intersect_hv_lines_through:NN { #3 } { #4 } }
42 { \_stridi_intersect_hv_lines_through:NN { #5 } { #6 } }
43 }
44 }
45 }

```

(End of definition for `_stridi_subdivide_segment:nNNNNN`.)

box Define a rectangular shape with configurable ports.

```

46 \pgfdeclareshape{box}{
47
48 % Inherit all the structure of rectangle
49 \inheritssavedanchors[from=rectangle]
50 \clist_map_inline:nn
51 {
52 north-west, north, north-east,
53 west, center, east,
54 mid-west, mid, mid-east,
55 base-west, base, base-east,
56 south-west, south, south-east,
57 }
58 { \inheritanchor[from=rectangle]{#1} }
59 \inheritanchorborder[from=rectangle]
60 \inheritbackgroundpath[from=rectangle]
61
62 % Dump port counts into saved macros
63 \savedmacro\portsnorth
64 { \pgfmathtruncatemacro\portsnorth{\pgfkeysvalueof{/pgf/box~ports~north}} }
65 \savedmacro\portseast
66 { \pgfmathtruncatemacro\portseast{\pgfkeysvalueof{/pgf/box~ports~east}} }
67 \savedmacro\portssouth
68 { \pgfmathtruncatemacro\portssouth{\pgfkeysvalueof{/pgf/box~ports~south}} }
69 \savedmacro\portswest
70 { \pgfmathtruncatemacro\portswest{\pgfkeysvalueof{/pgf/box~ports~west}} }
71
72 % Add ports definitions to shape definition
73 \expandafter\pgfutil@g@addto@macro\csname pgf@sh@s@box\endcsname{
74 \_stridi_subdivide_segment:nNNNNN { pgf@anchor@box@north } { \portsnorth }
75 { \southwest } { \northeast } { \northeast } { \northeast }
76 \_stridi_subdivide_segment:nNNNNN { pgf@anchor@box@east } { \portseast }
77 { \northeast } { \northeast } { \northeast } { \southwest }
78 \_stridi_subdivide_segment:nNNNNN { pgf@anchor@box@south } { \portssouth }
79 { \southwest } { \southwest } { \northeast } { \southwest }
80 \_stridi_subdivide_segment:nNNNNN { pgf@anchor@box@west } { \portswest }
81 { \southwest } { \northeast } { \southwest } { \southwest }

```

```

82   }
83
84 }

```

(End of definition for box. This function is documented on page ??.)

/pgf/box Define style to draw boxes.

```

85 \ExplSyntaxOff
86 \tikzset{
87   box/.default={0/0/0/0},
88   box/.style args={#1}{
89     shape=box,
90     draw,
91     inner sep=.5em,
92     minimum width=2em,
93     minimum height=2em,
94     execute at begin node=$,
95     execute at end node=$,
96     /pgf/box ports=#1,
97   },
98 }
99 \ExplSyntaxOn

```

(End of definition for /pgf/box. This function is documented on page 1.)

/pgf/dot Define style to draw dots.

```

100 \ExplSyntaxOff
101 \tikzset{
102   dot/.style={
103     shape=circle,
104     fill,
105     inner sep=0,
106     minimum width=0.4em,
107   },
108 }
109 \ExplSyntaxOn

```

(End of definition for /pgf/dot. This function is documented on page 3.)

\wires Define our main actor.

```

110 \NewDocumentCommand{\wires}{o m m }
111 {
112   \prop_set_from_keyval:Nn \l_tmpa_prop { #2 }
113   \prop_map_inline:Nn \l_tmpa_prop
114   {
115     \prop_set_from_keyval:Nn \l_tmpb_prop { ##2 }
116     \prop_map_inline:Nn \l_tmpb_prop
117     {
118       \regex_match_case:nnTF
119       {
120         { \. north } { \tl_gset:Nn \g_tmpa_tl { 90 } }

```

```

121     { \. south } { \tl_gset:Nn \g_tmpa_tl { -90 } }
122     { \. west } { \tl_gset:Nn \g_tmpa_tl { 180 } }
123     { \. east } { \tl_gset:Nn \g_tmpa_tl { 0 } }
124   } { #####2 } {} {}
125   \regex_match_case:nnTF
126   {
127     { north } { \tl_gset:Nn \g_tmpb_tl { 90 } }
128     { south } { \tl_gset:Nn \g_tmpb_tl { -90 } }
129     { west } { \tl_gset:Nn \g_tmpb_tl { 180 } }
130     { east } { \tl_gset:Nn \g_tmpb_tl { 0 } }
131   } { #####1 } {} {}
132   \draw [
133     out={\tl_use:N \g_tmpb_tl},
134     in={\tl_use:N \g_tmpa_tl},
135     #1,
136   ] (##1.#####1) to (#####2);
137 }
138 }
139 \clist_set:Nn \l_tmpa_clist { #3 }
140 \clist_map_inline:Nn \l_tmpa_clist {
141   \regex_match_case:nnTF
142   {
143     { \. north } { \draw[#1] (##1) -- +(0,+1); } % TODO: cleaner solution?
144     { \. south }
145     {
146       \draw[out=-90, in=0, #1] (##1)
147         to ($(\pgf@picminx, \pgf@y)$);
148     } % TODO: not sure why this works
149     { \. west } { \draw[#1] (##1) -- +(-1, 0); }
150     { \. east } { \draw[#1] (##1) -- +(1, 0); }
151   } { ##1 } {} {}
152 }
153 }

```

(End of definition for `\wires`. This function is documented on page 3.)

Close the DocStrip guards and call it a day.

```

154 \endpackage

```

Change History

0.1.0		<code>/pgf/box</code> : acts as a shortcut for	
General: initial version	1	setting port counts	6
0.2.0		<code>box</code> : make ports configurable through	
General: make box ports configurable	1	TikZ keys	5

Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

Symbols	
\.	120, 121, 122, 123, 143, 144, 149, 150
/pgf/box	1, <u>85</u>
/pgf/box_ports	2, <u>10</u>
/pgf/box_ports_east	2, <u>10</u>
/pgf/box_ports_north	2, <u>10</u>
/pgf/box_ports_south	2, <u>10</u>
/pgf/box_ports_west	2, <u>10</u>
/pgf/dot	3, <u>100</u>
B	
box	<u>46</u>
C	
clist commands:	
\clist_map_inline:Nn	140
\clist_map_inline:nn	50
\clist_set:Nn	139
\l_tmpa_clist	139, 140
cs commands:	
\cs_if_exist:NTF	29
\cs_new:Nn	22, 27
\cs_new_nopar:Npn	33
\csname	73
D	
\draw	132, 143, 146, 149, 150
E	
\endcsname	73
\expandafter	73
\ExplSyntaxOff	85, 100
\ExplSyntaxOn	99, 109
I	
\inheritanchor	58
\inheritanchorborder	59
\inheritbackgroundpath	60
\inheritsavedanchors	49
int commands:	
\int_step_inline:nnnn	28
N	
\NewDocumentCommand	110
\northeast	75, 77, 79, 81
P	
\pgfdeclareshape	46
\pgfextractx	23
\pgfextracty	24
\pgfkeys	10
\pgfkeysvalueof	64, 66, 68, 70
\pgfmathdivide	36
\pgfmathresult	40
\pgfmathtruncatemacro	64, 66, 68, 70
\pgfpoint	25
\pgfpointlineattime	39
\portseast	65, 66, 76
\portsnorth	63, 64, 74
\portssouth	67, 68, 78
\portswest	69, 70, 80
prg commands:	
\prg_break:	31
\prg_do_nothing:	32
prop commands:	
\prop_map_inline:Nn	113, 116
\prop_set_from_keyval:Nn	112, 115
\l_tmpa_prop	112, 113
\l_tmpb_prop	115, 116
\ProvidesExplPackage	5
R	
regex commands:	
\regex_match_case:nnTF	118, 125, 141
\RequirePackage	3, 4
S	
\savedmacro	63, 65, 67, 69
\southwest	75, 77, 79, 81
stridi internal commands:	
__stridi_intersect_hv_lines_-	
through:NN	22, 22, 41, 42
__stridi_subdivide_segment:nnnnnn	
.	27, 27, 74, 76, 78, 80
T	
T _E X and L ^A T _E X 2 _ε commands:	
\pgf@picminx	147
\pgf@xa	23, 25
\pgf@y	147
\pgf@ya	24, 25
\pgfutil@g@addto@macro	73
\tikzset	86, 101
tl commands:	
\tl_gset:Nn	120, 121, 122, 123, 127, 128, 129, 130

\tl_use:N	133, 134	W
\g_tmpa_tl	120, 121, 122, 123, 134	
\g_tmpb_tl	127, 128, 129, 130, 133	\wires 3, <u>110</u>