RULES

HANS HAGEN
A CONTEXT MKIV MANUAL
Introduction

This manual describes just one type of rules: those that somehow magically are bound to the typeset text. We will mention a few mechanisms that relate to this in the sense that they share some of the underlaying code and logic. The term “rules” should be interpreted liberally as we can kick in some MetaPost which then can get us away from straight rules.

This manual will not be that extensive, after all these mechanisms are not that complex to configure.

Underlining and overstriking

Already in ConTExt MkII we had underlining available but with some limitations. We could handle more than one word but at some point you hit the limits of the engine. The MkIV implementation is more flexible. In fact you can underline a whole document (which was actually a request from a user). This feature was also used by a colleague who was experimenting with texts for dyslectic readers.

This mechanism is generic in the sense that a framework is provided to define rules that run alongside text. Take this:

\underbars {drawing \underbar{bars} under words is a typewriter leftover}
\overstrikes {striking words makes them \overstrike {unreadable} but
sometimes even \overbar {top lines} come into view.}

This shows up as:

drawing bars under words is a typewriter leftover striking words makes them unreadable but sometimes even top lines come into view.

We can best explain what happens by looking at how these commands are defined:

\definebar[overbar] [method=1,dy=0.4, offset=1.8, continue=yes]
\definebar[underbar] [method=1,dy=-0.4,offset=-0.3,continue=yes]
\definebar[overstrike] [method=0,dy=0.4, offset=0.5, continue=yes]
\definebar[understrike] [method=0,offset=1.375, rulethickness=2.5, continue=yes, order=background]
\definebar[overbars] [overbar] [continue=no]
\definebar[underbars] [underbar] [continue=no]
\definebar[overstrikes] [overstrike] [continue=no]
\definebar[understrikes] [understrike] [continue=no]

The formal definitions of the commands are show in definition 1 and 2.
\definebar [...] [...][,...,...]

1. NAME
2. NAME
3. inherits: \setupbar

#### Definition 1 \definebar

\setupbar [...] [...][,...,...]

1. NAME

- \texttt{color} = \texttt{COLOR}
- \texttt{continue} = \texttt{yes no all}
- \texttt{empty} = \texttt{yes no}
- \texttt{unit} = \texttt{ex em pt in cm mm sp bp pc dd cc nc}
- \texttt{order} = \texttt{foreground background}
- \texttt{rulethickness} = \texttt{DIMENSION}
- \texttt{method} = \texttt{NUMBER}
- \texttt{offset} = \texttt{NUMBER}
- \texttt{dy} = \texttt{NUMBER}
- \texttt{max} = \texttt{NUMBER}
- \texttt{foregroundstyle} = \texttt{STYLE COMMAND}
- \texttt{foregroundcolor} = \texttt{COLOR}
- \texttt{mp} = \texttt{NAME}
- \texttt{left} = \texttt{TEXT}
- \texttt{right} = \texttt{TEXT}

#### Definition 2 \setupbar

The dy parameter specifies the shift up or down. The offset defines how nested bars are shifted.

The method determines centering of the bar: we set it to zero when we want an overstrike.

The continue parameter is responsible for drawing over spaces and the order determines the layering.

The units are either hard coded values like points or relate to the font at the spot of the bar. Here are some examples:

\setupbars[unit=mm,rulethickness=1] \quad bar \underbar{foo} \quad bar\quad\quad
\setupbars[unit=ex,rulethickness=1] \quad bar \underbar{foo} \quad bar\quad\quad
\setupbars[unit=pt,rulethickness=1] \quad bar \underbar{foo} \quad bar\quad\quad
\setupbars[unit=pt,rulethickness=10pt] \quad bar \underbar{foo} \quad bar

As if underlining wasn’t already bad enough, of course at some point there came a request for dashed lines.

\test \underrandoms{test me} and \underrandom{test} or \underrandom{grep}
\test \underdashes {test me} and \underdash {test} or \underdash {grep}
\test \underdots {test me} and \underdot {test} or \underdot {grep}

The above variants are predefined and render as:
A graphic is defined as follows. It boils down to drawing one or more shapes. In this example we also force a specific bounding box so that the result gets positioned right.

\startuseMPgraphic{rules:under:...}
\draw
\((0,\text{RuleDepth}) -- (\text{RuleWidth},\text{RuleDepth})\)
\shifted (0,\text{RuleFactor}*\text{RuleOffset})
\withpen pencircle scaled \text{RuleThickness}
\withcolor \text{RuleColor} ;
\setbounds currentpicture to unitsquare xysized(\text{RuleWidth},\text{RuleHeight}) ;
\stopuseMPgraphic

The following variables are available:

<table>
<thead>
<tr>
<th>variable</th>
<th>type</th>
<th>meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>RuleDirection</td>
<td>string</td>
<td>the direction of the line</td>
</tr>
<tr>
<td>RuleOption</td>
<td>string</td>
<td>whatever the caller finds useful</td>
</tr>
<tr>
<td>RuleWidth</td>
<td>number</td>
<td>the requested width of the rule</td>
</tr>
<tr>
<td>RuleHeight</td>
<td>number</td>
<td>the requested height of the rule</td>
</tr>
<tr>
<td>RuleDepth</td>
<td>number</td>
<td>the requested depth of the rule</td>
</tr>
<tr>
<td>RuleThickness</td>
<td>number</td>
<td>the linewidth</td>
</tr>
<tr>
<td>RuleFactor</td>
<td>number</td>
<td>the set factor (e.g. an ex)</td>
</tr>
<tr>
<td>RuleOffset</td>
<td>number</td>
<td>an (optional) offset in case of nesting</td>
</tr>
<tr>
<td>RuleColor</td>
<td>color</td>
<td>the color</td>
</tr>
</tbody>
</table>

The RuleFactor can be used as multiplier for the RuleOffset. Later we will see an example of how to use the RuleDirection and RuleOption.

The extra under commands are defined as follows. Watch the mp parameter: it refers to a graphic.

\definebar
[undergraphic]
[mp=rules:under:dash, offset=-.2, order=background]
\definebar[underrandom] [undergraphic][mp=rules:under:random]
\definebar[underdashes][underrandom][continue=yes]
\definebar[underdash] [undergraphic][mp=rules:under:dash]
\definebar[underdashes][underdash][continue=yes]
\definebar[underdot] [undergraphic][mp=rules:under:dots]
\definebar[underdots][underdot][continue=yes]

A nasty side effect of the implementation is that because we look mostly at glyphs, optionally separated by glue or kern some text might get unseen and therefore not treated.
A running box is seen as text. As you (probably) expect, a nested ornamental rule is supported as well:

\definebar
[xbartwo]
\text={\lower\exheight\hbox{\darkblue\infofont +},
repeat=yes}
\definebar
[xbarone]
\text={\lower\exheight\hbox{\darkred\infofont +},
repeat=yes}

Klein : \xbarone{\samplefile{klein}\removeunwantedspaces}\par
Sapolsky : \xbartwo{\samplefile{sapolsky}\removeunwantedspaces}\par

Klein : We don’t go into a state of shock when something big and bad happens; it has to be something big and bad that we do not yet understand. A state of shock is what results when a gap opens up between events and our initial ability to explain them. When we find ourselves in that position, without a story, without our moorings, a great many people become vulnerable to authority figures telling us to fear one another and relinquish our rights for the greater good.

Sapolsky : Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world.
Shifting

We mention shifting here because it shares code with the bars. There are two shifts defined but you can define more:

\defineshift
 [shiftup]
 [method=0,  
  dy=-1,  
  unit=ex,  
  continue=yes,  
  style=\txtx]  

\defineshift
 [shiftdown]
 [method=1,  
  dy=.3,  
  unit=ex,  
  continue=yes,  
  style=\txtx,  
  color=]  

An example of using the commands defined this way is:

Let's go \texttt{\shiftup{up}} and \texttt{\shiftdown{down}} a bit!

or: Let's go \texttt{up} and \texttt{down} a bit! Here we just shift words but you can shift more than that although I haven't yet seen a useful example of that:

We can \texttt{\shiftup {\samplefile{tufte}}} whole paragraphs if we really want.

We can thrive in information-thick worlds because of our marvelous and everyday capacity to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthesize, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeonhole, pick over, sort, integrate, blend, inspect, filter, lump, skip, smooth, chunk, average, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsize, winnow the wheat from the chaff and separate the sheep from the goats. whole paragraphs if we really want.

The formal definitions are given in definition 3, 4 and 5. The align switch is there for directional (and testing) purposes and is normally not used (or even useful in a line). The dy is multiplied by the factor that itself can depend on the used font.

\begin{verbatim}
\defineshift [\ldots] [\ldots] [\ldots]\texttt{\OPT \OPT}
\OPT 1 \texttt{NAME}
\OPT 2 \texttt{NAME}
\OPT 3 \texttt{inherits: \setupshift}
\end{verbatim}

\textbf{Definition 3} \texttt{\defineshift}
Fillers

The possibility of line fillers was mentioned by Mojca on the ConTExt mailing list and it’s actually not that hard to implement them. The only need I ever had for it was to fill out lines on some legal form and that was actually just some fun challenge in MkII times. The code got lost and never made it into ConTExt. This time it was added as a side effect of a thread at the tenth ConTExt meeting.

The ideas is to fill the rest of a line with some kind of (ornamental) rule. I’m not sure what sense it makes, even in legal documents. If it is to prevent additions then one should wonder if additions at the end of a (kind of arbitrary) broken line is what we should be afraid of most. So, for now, let’s consider it an educational feature.

\begin{verbatim}
\definelinefiller
  [filler-1]
  [height=.75\exheight, 
    distance=.25\emwidth, 
    rulethickness=.25\exheight, 
    textcolor=darkyellow, 
    before=\blank, 
    after=\blank, 
    color=darkred]

\startlinefiller[filler-1]
\samplefile{ward}
\stoplinefiller
\end{verbatim}

Here we define a filler. As you can see, a rule gets added at the end of a paragraph.
effect of an old-age patient who smokes many packs of cigarettes per day—and we humans are the cigarettes.

This time we don’t justify:

\startalign[flushleft,broad]
\startlinefiller[filler-1]
\samplefile{ward}
\stoplinefiller
\stopalign

Now more lines get a rule appended:

The Earth, as a habitat for animal life, is in old age and has a fatal illness. Several, in fact. It would be happening whether humans had ever evolved or not. But our presence is like the effect of an old-age patient who smokes many packs of cigarettes per day—and we humans are the cigarettes.

Before we continue it must be noted that the environment creates a paragraph. If you don’t want that you need to use \setlinefiller instead. Next we show a middle alignment:

\startalign[middle]
\startnarrower
\startlinefiller[filler-1]
\samplefile{ward}
\stoplinefiller
\stopnarrower
\stopalign

Here we get:

The Earth, as a habitat for animal life, is in old age and has a fatal illness. Several, in fact. It would be happening whether humans had ever evolved or not. But our presence is like the effect of an old-age patient who smokes many packs of cigarettes per day—and we humans are the cigarettes.

Let’s add another level of complexity, left- and right skips:

\startalign[middle]
\startnarrower
\startlinefiller[filler-1][scope=global]
\samplefile{ward}
\stoplinefiller
\stopnarrower
\stopalign

Here we get:

The Earth, as a habitat for animal life, is in old age and has a fatal illness. Several, in fact. It would be happening whether humans had ever evolved or not. But our presence is like the effect of an old-age patient who smokes many packs of cigarettes per day—and we humans are the cigarettes.

The lines stay within the narrower boundaries but you can extend them to the margins if you like:

\startalign[middle]
\startnarrower
\startlinefiller[filler-1][scope=global]
\samplefile{ward}
\stoplinefiller
\stopnarrower
The Earth, as a habitat for animal life, is in old age and has a fatal illness. Several, in fact. It would be happening whether humans had ever evolved or not. But our presence is like the effect of an old-age patient who smokes many packs of cigarettes per day—and we humans are the cigarettes.

You can also use a left or right scope, as in:

\startalign[middle]
  \startnarrower
    \startlinefiller[filler-1][scope=right]
      \samplefile{ward}
    \stoplinefiller
  \stopnarrower
\stopalign

Only the right rules extend into the margins.

So:

The Earth, as a habitat for animal life, is in old age and has a fatal illness. Several, in fact. It would be happening whether humans had ever evolved or not. But our presence is like the effect of an old-age patient who smokes many packs of cigarettes per day—and we humans are the cigarettes.

Of course these rules are somewhat boring so let’s now kick in some MetaPost.
The previous example now looks like:

The Earth, as a habitat for animal life, is in old age and has a fatal illness. Several, in fact. It would be happening whether humans had ever evolved or not. But our presence is like the effect of an old-age patient who smokes many packs of cigarettes per day—and we humans are the cigarettes.

This time we also change the direction and we can let the MetaPost graphic adapt to that by reverting the arrows.

The direction TRT is \TeX{} speak for a right-to-left direction. We use a latin script example for convenience.

Here we have adapted the graphic a bit:

```plaintext
if RuleDirection == "TRT" : reverse fi
((0,RuleHeight) -- (RuleWidth,RuleHeight))
if RuleOption == "right" : reverse fi
```

((0,RuleHeight) -- (RuleWidth,RuleHeight))
\definelinefiller \[\ldots\] \[\ldots\] \[\ldots\] 
\setuplinefiller \[\ldots\] \[\ldots\] \[\ldots\]

\begin{definition}
definelinefiller
\end{definition}

\begin{definition}
setuplinefiller
\end{definition}

\section*{User rules}

Characters and rules are the only graphical elements that \LaTeX{} really knows about. Even if you see images in a document, you should realize that they are just blobs with dimensions and that the backend replaces such blobs by real images.

The primitive operations for rules are \texttt{\textbackslash hrule} and \texttt{\textbackslash vrule} and the main difference is to what way they adapt to their situation when no dimensions are given and the mode change they trigger.

\texttt{hrule\{}\texttt{\textcolor{darkred}{\leaders\hrule height 1mm\relax\hfill}}\texttt{hrule}\par
\texttt{vrule\{}\texttt{\textcolor{darkyellow}{\leaders\vrule height 3mm depth 2mm}}\texttt{\relax\hfill}}\par
\texttt{hrule\{}\texttt{\textcolor{darkred}{\leaders\hrule width 10cm height 3mm depth 2mm}}\texttt{\relax\hfill}}\texttt{hrule}\par
\texttt{vrule\{}\texttt{\textcolor{darkyellow}{\leaders\vrule width 10cm height 3mm depth 2mm}}\texttt{\relax\hfill}}\texttt{vrule}\par

When more text is to follow you should end a specification with \texttt{\relax} to make sure that the scanner stops looking for more arguments. With \texttt{\leaders} you can create flexible rules.
In ConTeXt we also have so called frame rules:

\color[darkred]{\frule
  width 10cm
  height 1cm
  line 1mm
\relax}

This will produce a rectangle:

\hbox{some handy word with frames}\hfill

There are a few more keywords. Keep in mind that we actually have a new kind of primitive here, so we follow the TeX conventions of keywords.

\ruledhbox\bgroup
  \darkgray
  \frule width 100mm height 10mm depth 8mm radius 2mm line 2pt type fill\relax
  \hskip-100mm
  \darkred
  \frule width 100mm height 10mm depth 8mm radius 2mm line 2pt\relax
  \hskip-100mm
  \hbox to 100mm{\white \bold \hfill some handy word with frames\hfill}\%
\egroup

Of course this is a rather low level way of doing frames and such, but when you like that kind of low level programming you get the possibility here.

You can combine this with existing mechanisms. Take the following:

\defineoverlay[normalframe]
[\frule
  width \overlaywidth
  height\overlayheight
  line \overlaylinewidth
]
This is a variant on the already available round corners:

```
\defineoverlay[ovalframe]
[\frule
  width \overlaywidth
  height \overlayheight
  line \overlaylinewidth
  radius \overlayradius
]
```

The above result is accomplished with:

```
\hbox \bgroup
  \framed {test}
  \framed[frame=off] {test}
  \framed[background=normalframe,frame=off] {test}
  \framed[background=normalframe,frame=off] {test}
  \framed[corner=round] {test}
  \framed[corner=round] {test}
  \framed[background=ovalframe,frame=off] {test}
  \framed[background=ovalframe,frame=off] {test}
  \framed[background=ovalframe,frame=on] {test}
  \framed[background=ovalframe,frame=on] {test}
\egroup
```

Given the examples in the previous sections it will be no surprise that we can also use MetaPost.

```
\startuseMPgraphic{demoshape:back}
  fill
  unitcircle xysized (RuleWidth,RuleHeight+RuleDepth)
  withcolor RuleColor ;
\stopuseMPgraphic

\startuseMPgraphic{demoshape:fore}
  draw
  unitcircle xysized (RuleWidth,RuleHeight+RuleDepth)
  withcolor RuleColor
  withpen pencircle scaled 4RuleThickness ;
\stopuseMPgraphic
```

```
\hbox \bgroup
  \darkgray \frule width 100mm height 10mm depth 8mm type mp line 2pt
data \{\includeMPgraphic{demoshape:back}\}
\relax
  \hskip-100mm
  \darkred \frule width 100mm height 10mm depth 8mm type mp line 2pt
data \{\includeMPgraphic{demoshape:fore}\}
\relax
```
some handy word with frames

The \blackrule command is the more high level way to inject a rule.

\blackrule
     [width=10cm,
          height=1cm,
          depth=1cm,
          color=darkred]

This produces a boring rule:

Again we can revert to MetaPost:

\blackrule
     [width=10cm,
          height=1cm,
          depth=1cm,
          color=darkred,
          type=mp,
          mp=demoshape:back]

or:

The formal definition of this command is shown in definition 8 and 9.

**Hiding**

In education a to be filled in text is often represented by a gap in the running text and the bar drawing mechanism supports this. THere is a predefined \hiddenbar command:

\definebar
The Earth, as a habitat for animal life, is in old age and has a fatal illness. Several, in fact. It would be happening whether humans had ever evolved or not. But our presence is like the effect of an old-age patient who smokes many packs of cigarettes per day—and we humans are the cigarettes. 

The Earth, as a habitat for animal life, is in old age and has a fatal illness. Several, in fact. It would be happening whether humans had ever evolved or not. But our presence is like the effect of an old-age patient who smokes many packs of cigarettes per day—and we humans are the cigarettes. 

The Earth, as a habitat for animal life, is in old age and has a fatal illness. Several, in fact. It would be happening whether humans had ever evolved or not. But our presence is like the effect of an old-age patient who smokes many packs of cigarettes per day—and we humans are the cigarettes. 

The Earth, as a habitat for animal life, is in old age and has a fatal illness. Several, in fact. It would be happening whether humans had ever evolved or not. But our presence is like the effect of an old-age patient who smokes many packs of cigarettes per day—and we humans are the cigarettes.
Here is a variant that inserts spacing at the left and right edges. In this case the spacing is kept at a linebreak:

\definebar
  [widehiddenbar]
  [hiddenbar]
  [left={\quads[3]},
   right={\quads[3]}]

\widehiddenbar{invisible} \samplefile{weisman}
\widehiddenbar{invisible} \samplefile{weisman}
\widehiddenbar{invisible}

Since the mid-1990s, humans have taken an unprecedented step in Earthly annals by introducing not just exotic flora or fauna from one ecosystem into another, but actually inserting exotic genes into the operating systems of individual plants and animals, where they’re intended to do exactly the same thing: copy themselves, over and over.

Tabulate

The previously discussed mechanism is also available in the tabulate mechanism. We start with simple backgrounds:

\starttabulate
  \NL[darkred] foo \NC bar \NC \NR
  \NL[darkgreen] foo \NC bar \NC \NR
  \NL[darkblue] foo \NC \samplefile{tufte} \NC \NR
  \NL[darkgray] foo \NC bar \NC \NR
  \NL[darkyellow] foo \NC bar \NC \NR
\stoptabulate

This comes out as:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>foo</td>
<td>bar</td>
</tr>
<tr>
<td>foo</td>
<td>bar</td>
</tr>
<tr>
<td>foo</td>
<td>We thrive in information-thick worlds because of our marvelous and everyday capacity to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthesize, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeonhole, pick over, sort, integrate, blend, inspect, filter, lump, skip, smooth, chunk, average, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsize, winnow the wheat from the chaff and separate the sheep from the goats.</td>
</tr>
<tr>
<td>foo</td>
<td>bar</td>
</tr>
<tr>
<td>foo</td>
<td>bar</td>
</tr>
</tbody>
</table>
There are several two character commands that deal with this:

<table>
<thead>
<tr>
<th>command</th>
<th>related to effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>\ND</td>
<td>Normal with Default Line</td>
</tr>
<tr>
<td>\LB</td>
<td>Bold with Line</td>
</tr>
<tr>
<td>\DB</td>
<td>Bold with Default Line</td>
</tr>
<tr>
<td>\NF</td>
<td>Normal with Filler</td>
</tr>
<tr>
<td>\NP</td>
<td>Normal with Predefined Filler</td>
</tr>
<tr>
<td>\FB</td>
<td>Bold with Filler</td>
</tr>
<tr>
<td>\NA</td>
<td>Normal with Auto Toggled Line</td>
</tr>
<tr>
<td>\BA</td>
<td>Bold with Auto Toggled Line</td>
</tr>
</tbody>
</table>

Before we show more, we set up tabulate:

\setuptabulate
   \[blank={small,samepage},
   \headstyle=bold,
   \rulecolor=darkred,
   \rulethickness=1pt,
   \background=foo,
   \backgroundcolor=darkred,
   \foregroundcolor=white\]

This time we don’t set colors in the table itself:

\starttabulate[||l||]
\DB foos \BC bars \BC \NR
\TB
\NC foo foo foo \NC bar bar bar bar \NC \NR
\NC foo foo foo \NC bar bar bar bar \NC \NR
\NC foo foo foo \NC bar bar bar bar \NC \NR
\NC foo foo foo \NC bar bar bar bar \NC \NR
\LL
\stoptabulate

\bboxx
\bboxy
\bboxz
\bboxw
\bboxh
\bboxd

Instead of coming up with a separate mechanism for hooking in MetaPost we use the linefiller mechanism. We use these graphics:

\startuseMPgraphic{foo}
fill unitsquare
   xyscaled (RuleWidth,RuleHeight+RuleDepth) enlarged (ExHeight/4,ExHeight/8)
   shifted (-ExHeight/8,ExHeight/16)
   withcolor RuleColor ;
\stopuseMPgraphic
\startuseMPgraphic{bar}
  fill unitsquare
  xyscaled (RuleWidth,RuleHeight+RuleDepth) enlarged (ExHeight/4,ExHeight/8)
  shifted (-ExHeight/8,ExHeight/16)
  randomized ExHeight
  withcolor RuleColor ;
\stopuseMPgraphic

With these fillers:

\definelinefiller[foo][mp=foo,color=darkgreen]
\definelinefiller[bar][mp=bar,color=darkred]

An example of usage is:

\linefillerhbox[foo] to 12cm{\hss\strut\white\bf FOO\hss} \blank
\linefillerhbox[bar] to 10cm{\hss\strut\white\bf BAR\hss} \blank
\linefillerhbox[foo] to 9cm{\hss\strut\white\bf FOO\hss} \blank
\linefillerhbox[bar] to 14cm{\hss\strut\white\bf BAR\hss}

\hline
\begin{tabular}{|c|c|c|}
\hline
  foo & bars & \\
\hline
  foo foo foo & bar bar bar bar & \\
  foo foo foo & bar bar bar bar & \\
  foo foo foo & bar bar bar bar & \\
  foo foo foo & bar bar bar bar & \\
\hline
\end{tabular}
\hline
\stoptabulate

We can rely on the default:

\starttabulate[|||]
\PB foo \BC bars \BC \\
\NC foo foo foo \NC bar bar bar bar \NC \\
\NC foo foo foo \NC bar bar bar bar \NC \\
\NC foo foo foo \NC bar bar bar bar \NC \\
\NC foo foo foo \NC bar bar bar bar \NC \\
\stoptabulate

\begin{tabular}{|c|c|c|}
\hline
  foo & bars & \\
\hline
  foo foo foo & bar bar bar bar & \\
  foo foo foo & bar bar bar bar & \\
  foo foo foo & bar bar bar bar & \\
  foo foo foo & bar bar bar bar & \\
\hline
\end{tabular}
\hline

or be explicit:

\starttabulate[|||]
\FB[bar] foos \BC bars \BC \\
\NC foo foo foo \NC bar bar bar bar \NC \\
\NC foo foo foo \NC bar bar bar bar \NC \\
\NC foo foo foo \NC bar bar bar bar \NC \\
\NC foo foo foo \NC bar bar bar bar \NC \\
\stoptabulate

\begin{tabular}{|c|c|c|}
\hline
  foo & bars & \\
\hline
  foo foo foo & bar bar bar bar & \\
  foo foo foo & bar bar bar bar & \\
  foo foo foo & bar bar bar bar & \\
  foo foo foo & bar bar bar bar & \\
\hline
\end{tabular}
The auto variants will switch between colors:

```plaintext
\setuptabulate[backgroundcolor:1=darkred]
\setuptabulate[backgroundcolor:2=darkgreen]
\setuptabulate[backgroundcolor:3=darkblue]
```

```
\starttabulate[|||]
\BA foo foo foo \BC bar bar bar bar \NC \NR
\BA foo foo foo \BC bar bar bar bar \NC \NR
\BA foo foo foo \BC bar bar bar bar \NC \NR
\BA foo foo foo \BC bar bar bar bar \NC \NR
\BA foo foo foo \BC bar bar bar bar \NC \NR
\BA foo foo foo \BC bar bar bar bar \NC \NR
\BA foo foo foo \BC bar bar bar bar \NC \NR
\BA foo foo foo \BC bar bar bar bar \NC \NR
\BA foo foo foo \BC bar bar bar bar \NC \NR
\BA foo foo foo \BC bar bar bar bar \NC \NR
\BA foo foo foo \BC bar bar bar bar \NC \NR
```

```
\stoptabulate
```

<table>
<thead>
<tr>
<th>foos</th>
<th>bars</th>
</tr>
</thead>
<tbody>
<tr>
<td>foo</td>
<td>foo</td>
</tr>
<tr>
<td>foo</td>
<td>foo</td>
</tr>
<tr>
<td>foo</td>
<td>foo</td>
</tr>
<tr>
<td>foo</td>
<td>foo</td>
</tr>
<tr>
<td>foo</td>
<td>foo</td>
</tr>
<tr>
<td>foo</td>
<td>foo</td>
</tr>
<tr>
<td>foo</td>
<td>foo</td>
</tr>
<tr>
<td>foo</td>
<td>foo</td>
</tr>
<tr>
<td>foo</td>
<td>foo</td>
</tr>
<tr>
<td>foo</td>
<td>foo</td>
</tr>
<tr>
<td>foo</td>
<td>foo</td>
</tr>
</tbody>
</table>

```