

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 underlying 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 ConT_EXt 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...] [2...] [...,3...,...]
```

```
1 NAME
```

```
2 NAME
```

```
3 inherits: \setupbar
```

Definition 1 \definebar

```
\setupbar [...,1...] [...,2...,...]
```

```
1 NAME
```

```
2 color          = COLOR
  continue       = yes no all
  unit           = ex em pt in cm mm sp bp pc dd cc nc
  order          = foreground background
  rulethickness  = DIMENSION
  method         = NUMBER
  offset         = NUMBER
  dy             = NUMBER
  max            = NUMBER
  foregroundstyle = STYLE COMMAND
  foregroundcolor = COLOR
  mp            = NAME
```

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

```
bar foo bar bar foo bar bar foo bar bar foo 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:

```
test test me and test or grep
```

test test me and test or grep
 test test me and test or grep

A graphic is defined as follows. It boils down to drawing one or more shapes. In this example we also force a specific boundingbox so that the result gets positioned right.

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

The following variables are available:

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

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[underrandoms][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.

```
\underbars{We see this \high{\tfxx ®} symbol \runninghbox to 1cm{\hss} often.}
\underbar {We see this \high{\tfxx ®} symbol \runninghbox to 1cm{\hss} often.}
```

This gives:

We see this ® symbol _____ often.
We see this ® symbol _____ often.

A running box is seen as text. As you (probably) expect, a nested ornamental rule is supported as well:

```
\underbars{We see this \high{\tfxx\underdot{®}} symbol \runninghbox to 1cm{\hss} often.}
\underbar {We see this \high{\tfxx\underdot{®}} symbol \runninghbox to 1cm{\hss} often.}
```

This time we get (you might need a magnifier to see it):

We see this ® symbol _____ often.
We see this ® symbol _____ often.

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=\ttx]
```

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

An example of using the commands defined this way is:

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

or: Let's go ^{up} and _{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 \shiftup {\input{tufte}} whole paragraphs if we really want.

We can ^{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, synopsise, 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.

```
\defineshift [...1] [...2] [...3...3...]
```

```
1 NAME
```

```
2 NAME
```

```
3 inherits: \setupshift
```

Definition 3 `\defineshift`

```
\setupshift [...1...][...2...]
```

```
1 NAME
```

```
2 continue = yes no
  unit     = ex em pt in cm mm sp bp pc dd cc nc
  method   = NUMBER
  dy       = NUMBER
  align    = inherits: \setupalign
  style    = STYLE COMMAND
  color    = COLOR
```

Definition 4 `\setupshift`

```
\startshift [...*] ... \stopshift
```

```
* NAME
```

Definition 5 `\startshift`

Fillers

The possibility of line fillers was mentioned by Mojca on the ConT_EXt 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 ConT_EXt. This time it was added as a side effect of a thread at the tenth ConT_EXt 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.

```
\definelinefiller
```

```
[filler-1]
[height=.75\exheight,
 distance=.25\emwidth,
 rulethickness=.25\exheight,
 textcolor=darkyellow,
 before=\blank,
 after=\blank,
 color=darkred]
```

```
\startlinefiller[filler-1]
  \input ward
\stoplinefiller
```





Here we define a filler. As you can see, a rule gets added at the end of a paragraph.

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 don't justify:

```
\startalign[flushleft,broad]
  \startlinefiller[filler-1]
    \input 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:

 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]
      \input 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]
      \input ward
    \stoplinefiller
  \stopnarrower
\stopalign

```

This 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.

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

```

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

```

Only the right rules extend into the margins.

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 get rid of the left rules:





```

\startalign[middle]
  \startnarrower
    \startlinefiller[filler-1][scope=right,location=right]
      \input ward
    \stoplinefiller
  \stopnarrower

```


`\stopalign`

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.

```
\setuplinefiller
  [filler-1]
  [mp=rules:filler:demo,
  %threshold=.25\emwidth,
  color=darkred]

\startuseMPgraphic{rules:filler:demo}
  drawarrow
    if RuleDirection == "TRT" : reverse fi
    ((0,RuleHeight) -- (RuleWidth,RuleHeight))
  withpen
    pencircle scaled RuleThickness
  withcolor
    if RuleOption == "left" : complemented fi RuleColor ;
  setbounds currentpicture to
    unitsquare xysized(RuleWidth,RuleHeight) ;
\stopuseMPgraphic
```





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.

```
\startalign[middle,r2l]
  \startnarrower[4*middle]
    \startlinefiller[filler-1] [scope=global]
      \input ward
    \stoplinefiller
  \stopnarrower
\stopalign
```

The direction TRT is T_EX speak for a right-to-left direction. We use a latin script example for convenience.

 lataf a sah dna ega dlo ni si ,efil lamina rof tatibah a sa ,htraE ehT 
 rehtehw gnineppah eb dluow tI .tcaf ni ,lareveS .ssenlli 

← ekil si ecneserp ruo tuB .ton ro devlove reve dah snamuh ←
 ← fo skcap ynam sekoms ohw tneitap ega-dlo na fo tceffe eht ←
 ← .setteragic eht era snamuh ew dna — yad rep setteragic ←

The next rendering shows what happens when we set `\parindent` and `\parfillskip` to an excessive value of 100pt.

→ ega dlo ni si ,efil lamina rof tatibah a sa ,htraE ehT ←
 → rehtehw gnineppah eb dluow tI .tcaf ni ,lareveS .ssenlli lataf a sah dna ←
 → fo tceffe eht ekil si ecneserp ruo tuB .ton ro devlove reve dah snamuh ←
 → yad rep setteragic fo skcap ynam sekoms ohw tneitap ega-dlo na ←
 → .setteragic eht era snamuh ew dna — ←

Here we have adapted the graphic a bit:

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

```
\definelinefiller [1...] [2...] [3...=...,...]
```

- 1 NAME
- 2 NAME
- 3 inherits: \setuplinefiller

Definition 6 `\definelinefiller`

```
\setuplinefiller [1...,...] [2...,...=...,...]
```

- 1 NAME
- 2

location	=	left right <u>both</u>
scope	=	left right <u>local</u> global
color	=	COLOR
height	=	DIMENSION
depth	=	DIMENSION
distance	=	DIMENSION
threshold	=	DIMENSION
rulethickness	=	DIMENSION
textstyle	=	STYLE COMMAND
textcolor	=	COLOR
align	=	inherits: \setupalign
mp	=	NAME
before	=	COMMAND
after	=	COMMAND

Definition 7 `\setuplinefiller`

User rules

Characters and rules are the only graphical elements that $\text{T}_{\text{E}}\text{X}$ 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 `\hrule` and `\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.

```
hrule{\darkred \hrule width 10cm height 3mm depth 2mm}\par
vrule{\darkyellow\vrule width 10cm height 3mm depth 2mm}\par
hrule{\darkred \hrule width 10cm } \par
vrule{\darkyellow\vrule height 3mm depth 2mm}\par
```

```
hrule{\darkred \leaders\hrule height 1mm\relax\hfill}hrule\par
```

When more text is to follow you should end a specification with `\relax` to make sure that the scanner stops looking for more arguments. With `\leaders` you can create flexible rules.

```
hrule
```



```
vrule
```



```
hrule
```



```
vrule|
```

```
hrule
```



```
hrule
```

In ConT_EXt we also have so called frame rules:

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

This will produce a rectangle:



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



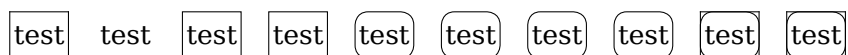
some handy word with frames

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

```
\defineoverlay[normalframe]
  [\frule
    width \overlaywidth
    height\overlayheight
    line \overlaylinewidth
  ]
```

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

This is a variant on the already available round corners:



The above result is accomplished with:

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

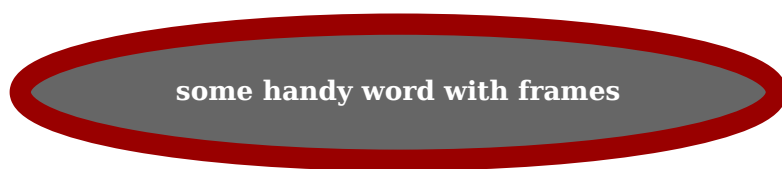
```

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
  \hskip-100mm
  \hbox to 100mm{\white \bold \hfill some handy word with frames\hfill}
\egroup

```

Or rendered:



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:

```
\setupblackrules [.,.,.*=.,.,.]

* width      = max DIMENSION
  height     = max DIMENSION
  depth      = max DIMENSION
  distance    = DIMENSION
  n           = NUMBER
  alternative = a b
  style       = STYLE COMMAND
  color       = COLOR
```

```
\blackrule [...=*...]
```

```
* inherits: \setupblackrules
```

Hiding

```
\definebar
[hiddenbar] [underbar]
[continue=yes,empty=yes,
left=\zwj,right=\zwj]
```

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. _____ 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. _____

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 previous text is generated with:

```
\input ward \hiddenbar                {\color[red]{invisible}}
\input ward \hiddenbar                {\quad\color[red]{invisible}\quad}
\input ward \hiddenbar{\quad\quad\quad\color[red]{invisible}\quad\quad\quad}
\input ward \hiddenbar                {\color[red]{invisible}\quad\quad\quad\quad\quad\quad}
\input ward
```

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} \input weisman
\widehiddenbar{invisible} \input 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. _____

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. _____