This Way

ConTEXt magazine #1105 LMTX
August 2020

Playing with boxes, a teaser

Hans Hagen

When a paragraph is typeset in a vertical box, we get a bunch of lines pasted together. If you want to change something in the result, you can disassemble that box in detailed ways (in Lua) but maybe it's something simple that you like to do. For that we have a an interface to the lines on such a box. We start with a few settings:

```
\parindentmode     \plusone
\normalizelinemode\plusone
```

These two settings will become default in ConTeXt LMTX, but here we need them in order to show some properties of boxes.

```
\setbox0\ruledvbox \bgroup
    \leftskip         10pt
    \rightskip        20pt
    \hangindent       40pt
    \hangafter         2
    \parindent         5pt
    \parfillleftskip  80pt plus 1 fill
    \parfillrightskip 40pt plus 1 fill
    \input tufte
\egroup
```

This box has a bit weird setup but we want to demonstrate something to it's okay:

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.

In case you wonder, the left and right parskips are a MkIV thing and supported in LMTX in a more advanced way. The left one can be used to push the last line to the right.

```
\dorecurse {\boxlines 0} {
    [#1] (%
        \the\boxlinewd 0 #1, \the\boxlineht 0 #1, \the\boxlinedp 0 #1,
        \the\boxlinels 0 #1, \the\boxliners 0 #1,
        \the\boxlinelh 0 #1, \the\boxlinerh 0 #1,
        \the\boxlinein 0 #1, \the\boxlinelp 0 #1, \the\boxlinerp 0 #1
    ) \par
}
```

```
[1] (423.94244pt, 7.276pt, 2.82845pt, 10.0pt, 20.0pt, 0.0pt, 0.0pt, 5.0pt, 0.0pt, 0.0pt)
[2] (423.94244pt, 7.25601pt, 2.82845pt, 10.0pt, 20.0pt, 0.0pt, 0.0pt, 0.0pt, 0.0pt, 0.0pt)
[3] (423.94244pt, 7.276pt, 2.82845pt, 10.0pt, 20.0pt, 40.0pt, 0.0pt, 0.0pt, 0.0pt, 0.0pt)
```

```
[4] (423.94244pt, 7.276pt, 2.82845pt, 10.0pt, 20.0pt, 40.0pt, 0.0pt, 0.0pt, 0.0pt, 0.0pt)
[5] (423.94244pt, 7.276pt, 2.82845pt, 10.0pt, 20.0pt, 40.0pt, 0.0pt, 0.0pt, 0.0pt, 0.0pt)
[6] (423.94244pt, 7.25601pt, 2.82845pt, 10.0pt, 20.0pt, 40.0pt, 0.0pt, 0.0pt, 0.0pt, 0.0pt)
[7] (423.94244pt, 7.276pt, 2.82845pt, 10.0pt, 20.0pt, 40.0pt, 0.0pt, 0.0pt, 0.0pt, 0.0pt)
[8] (423.94244pt, 7.276pt, 2.82845pt, 10.0pt, 20.0pt, 40.0pt, 0.0pt, 0.0pt, 0.0pt, 0.0pt)
[9] (423.94244pt, 7.276pt, 2.82845pt, 10.0pt, 20.0pt, 40.0pt, 0.0pt, 0.0pt, 131.69211pt, 91.69211pt)
```

The properties of a line are: width, height and depth, left and right skip, applied left and right hang, indentation (of the first line), left and right filler of the last line. You can use the commands shown as other dimensions, so `\the` is only needed when you serialize a property.

Some properties can be changed (others might follow later). Here we change the width of line four (the equal sign is optional):

```
\the\boxlinewd 0 4 \space& \boxlinewd 0 4 = 40pt \the\boxlinewd 0 4
```

We get: 423.94244pt & 40.0pt. We can detach a line from the box. Here we put two lines in another box that gets assigned to a box register, and a third line is assigned to a box register as-is:

```
\setbox2\ruledhbox{\copyboxline 0 2}
\setbox4\ruledhbox{\boxline     0 4}
\setbox6           \boxline     0 6
```

The result is:

```
[2] [\box2] \par
[4] [\box4] \par
[6] [\box6] \par
```

[2] [ity to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthe-]
[4] [] classify, list, abstract, scan, look into, idealize, isolate, discriminate, distinguish,
[6] [smooth, chunk, average, approximate, cluster, aggregate, outline, summarize,]

You can query the natural width, height and depth too:

```
used width  : \the\boxlinewd 0 5, natural width  : \the\boxlinenw 0 5
used heigth : \the\boxlineht 0 5, natural height : \the\boxlinenh 0 5
used depth  : \the\boxlinedp 0 5, natural depth  : \the\boxlinend 0 5
```

used width : 423.94244pt, natural width : 413.34448pt
used heigth : 7.276pt, natural height : 7.276pt
used depth : 2.82845pt, natural depth : 2.82845pt

Next we replace an existing line by a new line:

```
\setboxline 0 3 \hbox {\darkred \bold This is a replacement line!}
```

The original box is now:

We thrive in information–thick worlds because of our marvelous and everyday capacity to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthe-
This is a replacement line!

screen, pigeonhole, pick over, sort, integrate, blend, inspect, filter, lump, skip,

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.

We can repack the box with the usual TEX commands: `\vpack{\unvbox0}` but here we just did a `\box0`.

An example of a more useful application is the following. First we fill a box:

```
\setbox\scratchbox\ruledvbox \bgroup
    \input ward
\egroup
```

With little effort we can add line numbers:

```
\ruledvbox \bgroup
    \dorecurse {\boxlines \scratchbox} {%
        \dontleavehmode
        \strut
        \llap{\hbox to 1.5em{\txx#1\hss}}\boxline \scratchbox #1\par
    }
}
```

The result is:

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

source code of this document

```
% author    : Hans Hagen
% copyright : PRAGMA ADE & ConTeXt Development Team
% license   : Creative Commons Attribution ShareAlike 4.0 International
% reference : pragma-ade.nl | contextgarden.net | texlive (related) distributions
% origin    : the ConTeXt distribution
%
% comment   : Because this manual is distributed with TeX distributions it comes with a rather
%             liberal license. We try to adapt these documents to upgrades in the (sub)systems
%             that they describe. Using parts of the content otherwise can therefore conflict
%             with existing functionality and we cannot be held responsible for that. Many of
%             the manuals contain characteristic graphics and personal notes or examples that
%             make no sense when used out-of-context.

\ifdefined\parindentmode     \parindentmode    \plusone \fi % this will become default in lmtx
\ifdefined\normalizelinemode \normalizelinemode\plusone \fi % this will become default in lmtx

\usemodule[magazine-basic,abr-02]

\usemodule[scite]

\startbuffer[abstract]
    Here is yet another short example of a feature introduced in \CONTEXT\ \LMTX.
    It is mostly meant as teaser: maybe users have additional demands for this
    low level mechanisn.
\stopbuffer

\setuplayout
  [backspace=20mm,
   topspace=10mm]

\startdocument
  [title={Playing with boxes, a teaser},
   author={Hans Hagen},
  %affiliation=PRAGMA ADE,
   date=August 2020,
   number=1105 LMTX]

When a paragraph is typeset in a vertical box, we get a bunch of lines pasted
together. If you want to change something in the result, you can disassemble that
box in detailed ways (in \LUA) but maybe it's something simple that you like to
do. For that we have a an interface to the lines on such a box. We start with a
few settings:

\starttyping[option=TEX]
\parindentmode    \plusone
\normalizelinemode\plusone
\stoptyping

These two settings will become default in \CONTEXT\ \LMTX, but here we need them in
order to show some properties of boxes.

\startbuffer[sample]
\setbox0\ruledvbox \bgroup
    \leftskip         10pt
    \rightskip        20pt
    \hangindent       40pt
    \hangafter         2
    \parindent         5pt
    \parfillleftskip  80pt plus 1 fill
    \parfillrightskip 40pt plus 1 fill
    \input tufte
```

source code of this document

```
\egroup
\stopbuffer

\typebuffer[sample][option=TEX]

This box has a bit weird setup but we want to demonstrate something to it's okay:

\startlinecorrection
    \getbuffer[sample] \box0
\stoplinecorrection

In case you wonder, the left and right parskips are a \MKIV\ thing and supported
in \LMTX\ in a more advanced way. The left one can be used to push the last line
to the right.

\startbuffer[overview]
\dorecurse {\boxlines 0} {
    [#1] (%
        \the\boxlinewd 0 #1, \the\boxlineht 0 #1, \the\boxlinedp 0 #1,
        \the\boxlinels 0 #1, \the\boxliners 0 #1,
        \the\boxlinelh 0 #1, \the\boxlinerh 0 #1,
        \the\boxlinein 0 #1, \the\boxlinelp 0 #1, \the\boxlinerp 0 #1
    ) \par
}
\stopbuffer

\typebuffer[overview][option=TEX]

\getbuffer[sample]

\startpacked \tt \txx
    \getbuffer[overview]
\stoppacked

The properties of a line are: width, height and depth, left and right skip,
applied left and right hang, indentation (of the first line), left and right
filler of the last line. You can use the commands shown as other dimensions, so
\type {\the} is only needed when you serialize a property.

Some properties can be changed (others might follow later). Here we change the
width of line four (the equal sign is optional):

\startbuffer
\the\boxlinewd 0 4 \space& \boxlinewd 0 4 = 40pt \the\boxlinewd 0 4
\stopbuffer

\typebuffer[option=TEX]

We get: \inlinebuffer . We can detach a line from the box. Here we put two lines
in another box that gets assigned to a box register, and a third line is assigned
to a box register as|-|is:

\startbuffer
\setbox2\ruledhbox{\copyboxline 0 2}
\setbox4\ruledhbox{\boxline     0 4}
\setbox6           \boxline     0 6
\stopbuffer

\typebuffer[option=TEX] \getbuffer

The result is:

\startbuffer
```

source code of this document

```
[2] [\box2] \par
[4] [\box4] \par
[6] [\box6] \par
\stopbuffer

\typebuffer[option=TEX]

\startlinecorrection
    \startpacked
        \getbuffer
    \stoppacked
\stoplinecorrection

You can query the natural width, height and depth too:

\startbuffer
used width  : \the\boxlinewd 0 5, natural width  : \the\boxlinenw 0 5
used heigth : \the\boxlineht 0 5, natural height : \the\boxlinenh 0 5
used depth  : \the\boxlinedp 0 5, natural depth  : \the\boxlinend 0 5
\stopbuffer

\typebuffer[option=TEX] \startlines \getbuffer \stoplines

Next we replace an existing line by a new line:

\startbuffer
\setboxline 0 3 \hbox {\darkred \bold This is a replacement line!}
\stopbuffer

\typebuffer[option=TEX] \getbuffer

The original box is now:

\startlinecorrection
\box0
\stoplinecorrection

We can repack the box with the usual \TEX\ commands: \type {\vpack {\unvbox0}}
but here we just did a \type {\box0}.

An example of a more useful application is the following. First we fill a box:

\startbuffer
\setbox\scratchbox\ruledvbox \bgroup
    \input ward
\egroup
\stopbuffer

\typebuffer[option=TEX] \getbuffer

With little effort we can add line numbers:

\startbuffer
\ruledvbox \bgroup
    \dorecurse {\boxlines \scratchbox} {%
        \dontleavehmode
        \strut
        \llap{\hbox to 1.5em{\txx#1\hss}}\boxline \scratchbox #1\par
    }
}
\stopbuffer

\typebuffer[option=TEX]
```

source code of this document

```
The result is:

\startlinecorrection
    {\forgetall \getbuffer}
\stoplinecorrection

\stopdocument
```

source code of this document