

## Paweł Jackowski

### Custom overfull text

How to replace a black overfull rule at the end of too long lines of a paragraph?

Well, there is no a direct way to do so, but one should never underestimate T<sub>E</sub>X's bells and whistles. First of all, we can test if the last (h)box was overfull by checking the value of `\badness`; if it is larger than 10000 it definitely means that the box was overfull (`\badness` never exceeds 10000 for underfull boxes). Assuming that `\box0` is the box we want to test, we can say

```
\def\ooops{\hbox to\wd0{\setbox0=\hbox to\wd0{\unhbox0}%
\unhbox0 \ifnum\badness>10000 \rlap{\sevenrm\quad Ooops!}\fi}}
```

And how to get the box that is the line of a paragraph? By setting of the `\interlinepenalty` parameter to a strongly negative value we can force a page break between every two lines of a paragraph. In the `\output` routine, we can recognize those special penalties via the `\outputpenalty` parameter. The `\output` routine is not necessarily supposed to `\shipout` the page – it may simply return all its content back to the ‘recent contributions’.

```
\interlinepenalty=-50000 % force the break between each two lines
\maxdeadcycles=50 % allow upto 50 \outputs with no \shipout
\newtoks\orioutput \orioutput=\output % wrap the original \output routine
\output
{\ifnum\outputpenalty>-20000 \the\orioutput
\else \ifnum\outputpenalty<-\maxdimen \the\orioutput
\else
\unvbox255 % flush the entire list back
\setbox0=\lastbox % strip the very last box
\nointerlineskip % avoid doubled interline glue
\ooops % make the test and return the box back.
\advance\outputpenalty by50000
\penalty\outputpenalty % weak lie that nothing happened...
\fi\fi}
\hfuzz=\maxdimen % no overfullrule, no messages
\hsize=1.5in % provoke overfills
...
```

This completely useless example shows a not-so-useless trick, which `Ooops!` might be used for quite advanced applications, such as line-numbering, `Ooops!` some kind of paragraph decoration, page optimization and probably many others. Things become `Ooops!` much more complicated if math displays, `\marks`, `\inserts` or `\vad-` `Ooops!` `justs` come into play, but they don't spoil all of the game.

This completely useless example shows a not-so-useless trick, which `■` might be used for quite advanced applications, such as line-numbering, `■` some kind of paragraph decoration, page optimization and probably many others. Things become `■` much more complicated if math displays, `\marks`, `\inserts` or `\vad-` `■` `justs` come into play, but they don't spoil all of the game.