
\topmark in output routines without \shipout

Udo Wermuth

Abstract

Developers of formats for \TeX sometimes need to write their own output routine that replaces the macro `\plainoutput` of plain \TeX . One reason for a new output routine might be to do checks for a page without directly sending it to the dvi file. This article describes how one can avoid the `\topmark` of the next page becoming invalid in such a scenario.

1 Introduction

It is well-known that the program \TeX is accompanied by the file `plain.tex` that implements the format `plain`. The combination of program and format builds the system that the manual for \TeX , *The \TeX book* [3], explains. The contents of the file is also described in detail in the manual's Appendix B. A format is required as the program without a format, `INITEX`, provides only a minimal starting point and needs many additional declarations and definitions to become a useful tool for the average user. The file `plain.tex` delivers a set of more-or-less necessary additions.

The format `plain` provides not only a complete initialization of \TeX , it also includes macros that help users write common parts of various document types. For example, `plain` contains the macros `\beginsection` and `\proclaim`. The first outputs a heading and starts a section for a text; the second allows the presentation of a theorem in a mathematical paper. The macros of `plain` are usually efficient but limited, although they can handle a wide variety of texts. For example, a book project should replace the abovementioned macros. Donald E. Knuth describes in [3, App. E] the macros that he uses for *The \TeX book* and shows on page 259 a version of `\beginsection` for another of his book projects.

This observation is applicable to most macros of `plain`, even those that a user doesn't directly call, like the `\output` routine: `\output={\plainoutput}` [3, p. 255f.] where `\output` is a token list required by the program \TeX and `\plainoutput` a macro of `plain`. As part of a general purpose format these macros have certain limitations. Thus the \TeX user groups supported in \TeX 's early years the exchange of enhanced macros and later CTAN started to store them for general access. And the published papers provided feedback to \TeX 's manual: Knuth read the *TUGboat* articles until mid-1983 as a preparation for its Appendix D; see [5, comment after entry 775].

Udo Wermuth

doi.org/10.47397/tb/45-3/tb141wermuth-silent

One limitation of `\plainoutput` can be seen on this article's pages: It doesn't support multiple-column output; *TUGboat* provides its own output routine to get the two-column layout. (*The \TeX book* presents two methods to do that; see page 257 and pages 386–388.) Other problems show up when one has to work with spreads, i.e., the two pages that a reader sees when a book is opened. Publishers sometimes want to change the page height but keep it balanced for a spread; see [1]. (A remark about the reference: Nowadays one can query the badness of a created box via `\badness` and insertions can be kept using `\holdinginserts` [5, entries 883 and 884].)

Whenever `\plainoutput` is invoked it produces a page for the dvi file so that \TeX discards the processed data. But \TeX allows the output routine to return this data to the input stream for reprocessing. This behaviour is named here a *silent output path* if \TeX starts it because of a negative penalty.

Contents. Section 2 gives a brief overview about output routines without discussing the complex timing aspects and the associated lists. Section 3 lists a few use cases for reprocessing input. Section 4 discusses silent output paths and points out a problem with such output routines. Sections 5, 6, and 7 describe methods for how the problem can be solved.

2 Selected aspects of output routines

The program \TeX checks at several occasions, for example, at the end of a paragraph or after `\output` has ended [3, p. 122], if so much material was collected that the best page break (according to a *cost function* [3, p. 111]) was found. Usually that means more material than the page needs is available; otherwise the break might be forced because \TeX found a penalty ≤ -10000 . (Note, when \TeX finishes a very long paragraph it might have material for several pages; see also [3, ex. 14.15].) \TeX keeps all unused material for later processing. It is preceded by a `\penalty10000` if the break was at a penalty item. In this case \TeX stores the penalty of the breakpoint in the parameter `\outputpenalty`; otherwise `\outputpenalty` gets the value 10000 [3, p. 125].

As any penalty ≤ -10000 has the same effect, "reasons" for the call of `\output` can be coded with the values -10001 , -10002 , ... [3, p. 400]; `plain` uses the value -20000 to signal a `\supereject` that outputs all insertions that are on hold; see [3, pp. 254 and 353]. (-1073741824 is also special [3, p. 264].)

Box 255 and `\shipout`. Earlier the program \TeX removed from the top of the collected material all glue, kern, and penalty items and later it moves the

page’s main text to `\box255` without complaining about under- or overfull boxes [3, pp. 112 and 400]. `\box255` must be empty when communication with the output routine, the token list stored in `\output`, starts. \TeX fills `\box255` and the output routine must empty it again; `\plainoutput` does this via an `\unvbox255` [3, p. 120] inside a new vbox. In `\plainoutput`, a headline and footline are added and insertions are placed before this vbox is sent to the dvi file by the command `\shipout` [3, p. 254].

\TeX ’s integer parameter `\maxdeadcycles` determines after how many calls to `\output` it requires a `\shipout`. (INITEX sets `\maxdeadcycles = 25`.) Every time `\output` is called \TeX increases another parameter, `\deadcycles`, and resets its value to 0 after a `\shipout`. \TeX raises an error and ships out the page itself if `\deadcycles > \maxdeadcycles`. In this case the program \TeX executes something like “`\output={\shipout\box255}`” [3, p. 253].

Insertions and marks. The user’s `\output` routine receives not only `\box255`, which contains the text for the page, but also all boxes from the defined insertion classes [3, p. 122]. The data from each insertion class is cleared when it is transferred to its associated box for `\output`. This describes \TeX ’s default behaviour. The insertions remain in `\box255` if `\holdinginserts > 0` [3, p. 125].

`\topmark` receives the data from the `\botmark` of the previous `\box255`. `\firstmark` and `\botmark` are filled with the first and last mark of the current `\box255`. Without a `\mark` in this box, \TeX fills them with the contents of `\topmark`; see [3, p. 258].

Page data. With `\tracingpages > 0` \TeX documents the calculations of the cost function in the log [3, p. 112]. The trace shows the goal height and the current total together with the maximal stretch and shrink values for the page so far. One can inspect these values as \TeX makes them available as dimensions in pt, called `\pagegoal`, `\pagetotal` (the current height), `\pagedepth` (the current depth), `\pagestretch`, `\pagefilstretch` (in pt not fil), `\pagefillstretch` (in pt), `\pagefilllstretch` (in pt), and `\pageshrink` (see [3, pp. 114]). Some of them are used by `plain`’s `\midinsert` [3, p. 116] to check if the insertion fits on the current page or if it must be transformed into a delayed `\topinsert`.

`\pagegoal` is initialized with `\vsize` and decreased by insertions, i.e., when `\insert` is used. In `plain`, `\insert` is called by `\vfootnote`, which is called by `\footnote`, and by `\endinsert` if it closes a `\topinsert` or a transformed `\midinsert`.

Final remark. All actions inside `\output` are executed inside a group that the \TeX program adds.

Therefore, everything is a local change unless we prefix it by `\global` [3, p. 21].

3 Some reasons for a delayed `\shipout`

Let’s state two observations.

1. The `plain` format has only one insertion class for illustrations and figures: `\topins` [3, p. 363]; the `\endinsert` macro uses it. As the name indicates, the insertion is placed at the top of the page if there is enough room. Otherwise it is placed at the top of the next page that has enough space for this insertion after all previous `topinserts` have been placed. That is, the sequence of the insertions as they occur in the document is kept. This is not the case for `plain`’s `\midinsert`, as it isn’t handled as an insertion class. And a new insertion class, say `\botins`, would keep its own sequence independent of `\topins`. Thus, insertions might not appear in the sequence in which they occur in the input.

2. Figures 1 and 2 of [9] show that (a) the move of a club line from an odd-numbered page to the next and (b) the move of an insertion from a left page to the preceding spread improve the layout.

Therefore people want to control the placement of insertions themselves. Doing that automatically might best be handled in the output routine, with `\holdinginserts = 0`: Every time \TeX finds an insertion it is forced to call the output routine that decides about the placement even if there is not enough material for a complete page yet; that is, there will be no `\shipout`. As the input must not be lost but boxes must be emptied, the insertion is temporarily saved and in a later call to `\output` reprocessed.

Others want to change and balance the page height in a spread [1] or try to avoid widow and club (or orphan) lines at a page break [2], for example by changing the number of lines in a paragraph. Again, people look for automatic solutions of such problems and work with output routines that have paths without a `\shipout`.

4 Silent output paths

Page 254 of *The \TeX book* shows a do-nothing output routine. (I added the comments.)

```
\output={%a do-nothing output routine, TB p.254
\unvbox255 %      take page out of its \vbox
\ifnum\outputpenalty<10000 %      and reinsert
\penalty\outputpenalty % this penalty if
\fi}% there was a page break at a penalty item
```

The `\unvbox255` returns `\box255`’s contents to the input for reprocessing. If no parameters change, the same `\box255` is built when this or any other output routine is called. But we don’t have a chance to call another output routine if \TeX executes the do-

`\topmark` in output routines without `\shipout`

nothing output routine. (A reader who wants to test that will quickly get the abovementioned error message “Output loop—25 consecutive dead cycles”.) And remember every call to `\output` now adds a `\penalty10000` waiting to be processed by `TEX`. (Insert `\showlists` at the start of the do-nothing output routine to see this penalty item in your test.) Moreover, with `\holdinginserts = 0` insertions are lost from the input, i.e., they survive in their boxes but `TEX` does not change `\pagegoal` anymore for an `\output` with `\shipout`.

A silent output path. We want to call the silent output path in vertical mode via a unique negative penalty < -10000 and $\neq -20000$ that can be tested and deleted in `\output`. As the do-nothing routine only wastes resources let’s add something useful. The `\box255` is the last box that `TEX` creates before it fires up the `\output` routine; see [4, §1012 and §1017]. Thus, its badness is stored in `\badness` and can be accessed; see the remark in section 1.

```
\output={\ifnum\outputpenalty=-"CBAD % = -52141
\immediate\write16{Badness of page so far:
\the\badness}\unvbox255
\else\immediate\write16{Badness of completed
page: \the\badness}\plainoutput \fi}
```

The silent path of this output routine is triggered by a unique negative penalty: `-"CBAD`. To avoid `TEX`’s inserted `\penalty10000` cancelling page breaks, a penalty item should be used. It also ships out all completed pages so that only an unfinished page is left when the silent output path starts its work. Together with setting `\holdinginserts`, we put `\penalty0` into a macro; one can also move it out of the macro and then use other values < 10000 . The macro must be used only in vertical mode.

```
\def\dosilentoutput{% trigger silent output path
\edef\curmode{\ifinner-\fi\ifvmode 1 \else
\ifhmode 2 \else\ifmmode 3 \else 0 \fi\fi\fi}%
\ifnum\curmode=1 {\penalty0 % annihilates TeX’s
% \penalty10000; ships out all completed pages
\holdinginserts=1 \penalty-"CBAD }%
\else\errhelp{Call \dosilentoutput only in non%
-internal vertical mode. I’m forced to ignore
it.}\errmessage{Vertical mode required}\fi}
```

A side effect of silent output paths. In 1988 a `LATEX` user wrote about a *mysterious phenomenon*: “... when the marginal note instruction is added, the value of `\topmark` gets trashed.” Leslie Lamport answered [8]: “Not so mysterious... `LATEX` processes figures and marginal notes by calling the `\output` routine (by inserting a vertical penalty less than -10000). (This is done to enable `LATEX` to figure out whether the marginal note will be on an even- or odd-numbered page.) Of course, the value

of `\topmark` gets destroyed in the process.” (See also this newsletter (2022) <https://ctan.org/macros/latex/base/ltnews35.pdf> for new developments.)

The do-nothing output routine does nothing, as required, but the silent output path might destroy something if `\box255` contains a `\mark` command. It is correct that *The T_EXbook* writes that everything in `\box255` is returned for a second processing by `TEX`. But in section 2 it was stated that there is a link between the current page and the previous `\box255` via `\topmark`. When `TEX` builds a new `\box255` for a silent output path there might be a new token list for its `\botmark`, which becomes the `\topmark` for the next page, i.e., the next `\box255`.

An example. Let me illustrate the problem with an example. There are three pages and on each page there is a `\mark` containing the page number. Assume that a silent output path is triggered on page 3. Either the text contains the `\mark` after the silent output path was called or `TEX` saw the `\mark` when `\box255` is built for the silent output path. Thus we have two cases:

| page | mark | <code>\topmark</code> | <code>\firstmark</code> | <code>\botmark</code> |
|----------|------|-----------------------|-------------------------|-----------------------|
| [1] | 1 | null | 1 | 1 |
| [2] | 2 | 1 | 2 | 2 |
| 3 silent | -/3 | 2 | 2/3 | 2/3 |
| [3] | 3 | 2/3 | 3 | 3 |

and only the first case behaves as if the silent output path was not called. In the second case we see that the final third page does not have the correct `\topmark`, i.e., the contents of page 2’s `\botmark`.

Use the following code together with the above macro `\dosilentoutput` and the stated `\output` to see the problem for `\topmark`.

```
\headline{\hfil\tenrm % show marks in headline
\topmark---\firstmark---\botmark\hfil}
% three pages (with 2 lines) each with one \mark
\vsizetopskip \advance\vsizetopskip by \baselineskip
One\par \mark{1} Two\par % page 1
Three\par \mark{2} Four\par % page 2
Five\par \mark{3}\dosilentoutput Six\bye% page 3
Exchange \mark{3} and \dosilentoutput to get
the other case.
```

Final remark. As stated in section 2, at some point `TEX` throws away vertical skips, kerns, and penalties at the beginning of a page. Of course, a page should not begin, for example, with a vertical skip as we usually want to have the first lines of the pages at the same distance from the top edge of the paper. But this cancellation is not recoverable; whatever is thrown away at this place is gone forever. (See the conversation between Fred Bartlett, Donald E. Knuth, and Peter Breitenlohner

on page 20 of [7].) Thus, call a silent output path only at places where nothing is lost or where the loss of the mentioned items doesn't create a problem.

5 Keeping `\topmark` valid

There is only one way to get a valid `\topmark` after a silent output path is executed: A page must be built that contains the required mark contents as its last mark, i.e., as its `\botmark`. This must happen just before the page that has to be shipped out next is generated. Sure, the complete page that was shipped out last could be saved and reprocessed but this time without shipping it out so that everything is rebuilt. But actually only its `\botmark` is required.

To implement the above description we define three macros. The first stores the `\botmark` of a shipped-out page. As mentioned above a `\global` assignment is required as \TeX executes `\output` inside an (invisible) group.

```
\newbox\PrevPageBotMark
\def\SaveCurrentBotMark{% save the \botmark of a
% normal output routine, i.e., one with \shipout
\global\setbox\PrevPageBotMark% global required
=\vbox{\Restore\mark{\botmark}\vfil}}
\SaveCurrentBotMark %initialize \PrevPageBotMark
(Note that "Restore" or anything else that starts
horizontal mode is required to have later some ma-
terial for \box255; \vfil switches to vmode.)
```

The second macro must be called to finish all silent output paths. It creates a new page by unboxing the last stored `\botmark`. Next, it triggers another call to `\output` via a unique negative penalty ≤ -10000 .

```
\def\CloseSilent{% create page with the \botmark
% for silent output routine to keep the \topmark
\unvcopy\PrevPageBotMark \penalty-"COFF
\unvbox255 }
```

The third macro realizes the task of the second silent output path. It has to clear `\box255` that contains the newly created page of the above macro. As we want neither to ship out the page nor to unbox the page for reprocessing, `\box255` is unboxed into another box: I take `\box0`. This assignment is local and `\box0` is restored when `\output` ends.

```
\def\RestoreTopMark{% empty \box255, do not ship
% it out and do not put it back for reprocessing
\setbox0=\vbox{\unvbox255 }}
```

A corrected `\output`. The token list of the above `\output` gets a new case that tests the parameter `\outputpenalty` against `-"COFF` of the second macro.

```
\output={% this routine keeps the valid \topmark
\ifnum\outputpenalty=-"CBAD % = -52141
\immediate\write16{Badness of page so far:
```

```
\the\badness}\CloseSilent
\else\ifnum\outputpenalty=-"COFF % = -49407
\RestoreTopMark
\else\SaveCurrentBotMark
\immediate\write16{Badness of completed page:
\the\badness}\plainoutput
\fi\fi}
```

What happens with our example from the previous section? Now the table has one more line as there are two calls to silent output paths.

| page | mark | <code>\topmark</code> | <code>\firstmark</code> | <code>\botmark</code> |
|------------|------|-----------------------|-------------------------|-----------------------|
| [1] | 1 | null | 1 | 1 |
| [2] | 2 | 1 | 2 | 2 |
| 3 silent 1 | -/3 | 2 | 2/3 | 2/3 |
| 3 silent 2 | 2 | 2/3 | 2 | 2 |
| [3] | 3 | 2 | 3 | 3 |

Independent of whether there is a `\mark` on the page when the first silent output path gets called, the `\topmark` of the shipped-out page 3 is correct.

6 Using more than one output routine

In the previous section the silent output path was placed together with the default output routine inside the token list of `\output`. When we want to look at more than one page the silent output path should get its own `\output`. In this section we want to determine the net height of the box with the material that \TeX has collected for the next pages including insertions. (Not included are the stretch and shrink components of skips as well as the extra space of insertions ([3, p. 122]).) Thus, the value is different from `\pagetotal` that looks at a single page only.

Extended default. The first `\output` is based on `\plain`'s default. It must be extended by the procedure that reestablishes the `\topmark`. Again, we call this procedure with a negative penalty: `-"CODA`. As this is the point when the default output routine takes back control we also reset `\holdinginserts`. `\output={% plain \TeX 's output routine, extended % by two macros to keep \topmark valid`

```
\ifnum\outputpenalty=-"CODA % = -49370
\RestoreTopMark \global\holdinginserts=0
\else \SaveCurrentBotMark \plainoutput \fi}
```

We keep the declaration of `\PrevPageBotMark`, perform its initialization, and also reuse the two macros `\SaveCurrentBotMark` and `\RestoreTopMark` of the previous section.

Silent `\output`. The second output routine is activated through a macro. This silent output path is triggered by the negative penalty `-"CODE`. Then it displays the height of the box with all material that is waiting to be processed by \TeX for the next pages. We store the material in a box called `\SaveBoxCclv`.

`\topmark` in output routines without `\shipout`

```
\def\SilentOutput{% create a silent \output
\global\output={\TwoSilentPaths}}
```

The main task of the second output routine is done in the macro `\TwoSilentPaths`. In our example it is quite simple as it must only store material: (a) `\box255` and (b) if at a page break with a penalty, that penalty (i.e., the `\outputpenalty`).

```
\newbox\SaveBoxCclv
\def\TwoSilentPaths{%
\global\setbox\SaveBoxCclv % collect material
  =\vbox{\unvbox\SaveBoxCclv \unvbox255 }
\ifnum\outputpenalty=-"CODE % = -49374
\immediate\write16{height of box with the
  collected material: \the\ht\SaveBoxCclv}
\SwitchOutputRoutine % back to default output
\else\ifnum\outputpenalty<10000 % break at a
\global\setbox\SaveBoxCclv = \vbox{% penalty
\unvbox\SaveBoxCclv \penalty\outputpenalty}
\fi\fi}
```

Note that we need to replace `\CloseSilent` as the original output routine must be restored. And we need to `unvbox` not only `\PrevPageBotMark` but also the new box `\SaveBoxCclv`.

```
\def\SwitchOutputRoutine{%activate stored output
\global\output=\SaveOutput % restore \output
%end silent output routine and restore \topmark
\global\SilentOutputActivefalse
\unvcopy\PrevPageBotMark \penalty-"CODA
\unvbox\SaveBoxCclv\penalty0 }% return material
```

Activation. The last thing we have to do is to replace the macro `\dosilentoutput`. We need to store the current output routine in a new token list. Next we have to activate the above silent output path. Moreover, as we want to apply the silent output path to, for example, a very long paragraph we must split the macro. Otherwise the end of the paragraph triggers the extended output routine before the activation macro takes control. When two macros require to be called in sequence we should use error messages if a user calls them in the wrong order.

```
\newtoks\SaveOutput % save current \output
\newif\ifSilentOutputActive % changed \output?
\def\beginsilentoutput{% start the silent output
\ifSilentOutputActive
\errhelp{You already said \beginsilentoutput
and I'm using a silent \output routine. Maybe
an \endsilentoutput is missing. I'll forget
that you said \beginsilentoutput again.}%
\errmessage{Silent output routine is active}%
\else \global\SilentOutputActivetrue
\global\holdinginserts=1 % keep inserts
\global\SaveOutput=\output \SilentOutput
\fi}
```

The macro `\endsilentoutput` should only be called in vertical mode. Thus we write two error messages.

Udo Wermuth

```
\def\endsilentoutput{% activate the task of the
\ifSilentOutputActive % silent output routine
\ifvmode \penalty-"CODE
\else \errhelp{Call \endsilentoutput only in
vertical mode. I'm forced to ignore it.}%
\errmessage{Vertical mode required}\fi
\else\errhelp{You didn't say \beginsilentoutput
so I cannot end a silent \output routine. Go
on, nothing bad can happen.}\errmessage{No
silent output routine active}\fi}
\let\goodbye=\bye % forgotten \endsilentoutput?
\outer\def\bye{\ifSilentOutputActive \par
\immediate\write16{(end stops silent output)}%
\penalty-"CODE
\fi \csname goodbye\endcsname}
```

The last four lines prevent the output-loop error (see section 4) if `\endsilentoutput` was forgotten in the input.

An example. Let's take the example of section 4 and change it a little bit so that it becomes an example for this section.

```
\headline{\hfil\tenrm % show marks in headline
\topmark---\firstmark---\botmark\hfil}
% three pages (with 2 lines) each with one \mark
\vsizetopskip \advance\vsizetopskip by \baselineskip
One\par \mark{1} Two\par % page 1
Three\par \mark{2} Four\par % page 2
\beginsilentoutput Five\par \mark{3}
\endsilentoutput Six\bye % page 3
```

As expected all topmarks are correct. The message reports 32pt as we collect the text of page 2 (22pt), which was not yet output by \TeX when it sees the macro `\beginsilentoutput`, and the first line of page 3 (10pt).

Some remarks. In this section the prefix `\global` was applied in front of `\holdinginserts` (twice), `\output` (twice), and `\SaveOutput`. This allows, for example, to code `\beginsilentoutput` in a paragraph — together with `\endsilentoutput` after the `\par` — to call the silent output routine for this paragraph.

As before we use a `\penalty0` (see the macro `\SwitchOutputRoutine`) to defeat \TeX 's inserted `\penalty10000`. The contents of the shipped-out pages, shown with `\tracingoutput=2`, contain pairs of the form `(. . \penalty 0, . . \penalty 10000)` at places where the silent output path ends. For instance, page 3 of the above example shows such a pair where `\endsilentoutput` was called.

7 Combining both methods

The method of section 5 works for the last unfinished page, the one of section 6 for any material even if

TeX outputs it in more than one page. Here we show how one can use both methods in one document.

The sections start with different `\output` routines; the one of section 6 is the minimal requirement and we start with it although this makes the macros more complex as the macro `\dosilentoutput` of section 4 must activate the one of section 5.

```
\newif\ifOutputBadness
\def\ActivateBadnessReport{% for \dosilentoutput
\ifSilentOutputActive %("CBAD) in output routine
\else \OutputBadnesstrue % start report \badness
\output={%this routine keeps the valid \topmark
\ifnum\outputpenalty=-"CBAD % = -52141
\immediate\write16{Badness of page so far:
\the\badness}\CloseSilent
\else\ifnum\outputpenalty=-"CODA % = -49370
\RestoreTopMark \global\holdinginserts=0
\else \SaveCurrentBotMark
\immediate\write16{Badness of completed page:
\the\badness}\plainoutput
\fi\fi}\fi}
```

It's a combination of both output routines, for example, the test of `\outputpenalty` against `-"COFF` is gone and only the test `-"CODA` is used. It does not make sense to keep two identical tests. Note, we do not perform the activation between the macros `\beginsilentoutput` and `\endsilentoutput` since they change `\output`. To use `\dosilentoutput` between these macros, call `\ActivateBadnessReport` earlier than the macro `\beginsilentoutput`.

```
\def\dosilentoutput{% silent \badness reporting
\ifvmode %with preparation: both vertical modes
\ifOutputBadness\else\ActivateBadnessReport\fi
\ifOutputBadness \penalty0
\global\holdinginserts=1 \penalty-"CBAD
\else \errhelp{Call \ActivateBadnessReport
before \beginsilentoutput/ \endsilentoutput
if \dosilentoutput is used between them.}%
\errmessage{Cannot activate output routine}%
\fi
\else \errhelp{Call \dosilentoutput only in
vertical mode. I'll forget that I saw it.}%
\errmessage{Vertical mode required}\fi}
```

Next, we must change `\CloseSilent` of section 5 as it uses `-"COFF`. Moreover, in the case that it is executed between `\beginsilentoutput` and `\endsilentoutput` we cannot restore the box with the saved `\botmark` as it becomes a part of `\SaveBoxCclv`. Luckily, we don't need to do that.

```
\def\CloseSilent{% create page with the \botmark
% for silent output routine to keep the \topmark
\ifSilentOutputActive % except if called inside
\else \unvcopy\PrevPageBotMark % another silent
\fi \penalty-"CODA \unvbox255 }% output routine
```

The macro `\SilentOutput` must be extended to cover the silent output path of section 5.

```
\def\SilentOutput{% create a silent \output
\global\output={\ifnum\outputpenalty=-"CBAD
\immediate\write16{Badness of page so far:
\the\badness}\CloseSilent
\else \TwoSilentPaths
\fi}}
```

Acknowledgments. The topic “silent output path” occurred in a discussion between Max Chernoff, Karl Berry, and me. At the end there was an agreement that the results, although probably not new, should be documented to make them better known. I accepted the task to write a few pages for *TUGboat*. I used the contributions of the other two in this text without giving explicit credit. Careful proofreading by Karl Berry and Barbara Beeton improved readability.

References

- [1] Frederick H. Bartlett, “Automatic Page Balancing Macros Wanted”, *TUGboat* **9**:1, 1988, 83. tug.org/TUGboat/tb09-1/tb20bartlett.pdf
- [2] Max Chernoff, “Automatically removing widows and orphans with lua-widow-control”, *TUGboat* **43**:1, 2022, 28–39. tug.org/TUGboat/tb43-1/tb133chernoff-widows.pdf
Update in *TUGboat* **43**:3, 2022, 340–342. tug.org/TUGboat/tb43-3/tb135chernoff-lwc.pdf
- [3] Donald E. Knuth, *The TeXbook*, Volume A of *Computers & Typesetting*, Boston, Massachusetts: Addison-Wesley, 1984.
- [4] Donald E. Knuth, *TeX: The Program*, Volume B of *Computers & Typesetting*, Boston, Massachusetts: Addison-Wesley, 1986.
- [5] Donald E. Knuth, “The Errors of TeX”, *Software — Practice and Experience* **19** (1989), 607–685; reprinted as Chapters 10 and 11 in [6], 243–339. The log, i.e., Chapter 11, is still updated: ctan.org/tex-archive/systems/knuth/dist/errata/errorlog.tex
- [6] Donald E. Knuth, *Literate Programming*, CSLI Lecture Note No. 27, 1992. www-cs-faculty.stanford.edu/~knuth/lp.html
- [7] Donald E. Knuth, “TUG’95: Questions and Answers”, *TUGboat* **17**:1 (1996), 7–22. tug.org/TUGboat/tb17-1/tb50knut.pdf
- [8] Leslie Lamport, “LaTeX Notes”, *TeXhax Digest* **V88**:11, 1988-02-01; in: ctan.org/tex-archive/info/digests/texhax/88/texhax.11.gz
- [9] Michael F. Plass, *Optimal Pagination Techniques for Automatic Typesetting Systems*, Ph.D. thesis, Stanford University, 1981, 6+72 pages. tug.org/docs/plass/plass-thesis.pdf

◇ Udo Wermuth
Dietzenbach, Germany
[u dot wermuth \(at\) icloud dot com](mailto:u dot wermuth (at) icloud dot com)

`\topmark` in output routines without `\shipout`