Warnings

Solution to the puzzle from *TUGboat* 13#2: Where does this character come from?

Frank Mittelbach

Puzzle:

> If some complex macro defined by you produces funny extra characters like "Ω" or "œ" in the output, what kind of mistake could be the reason?

After the above puzzle was published several people sent me their solutions and they all thought of problems that I didn't have in mind when I was writing this short article.

Indeed, it is possible to sometimes get a weird character at the end of an input file, namely 'æ'. This is a control-Z (`^Z`), the old end-of-file marker from the DOS operating system. This character once marked the end of a file but is obsolete with newer versions of DOS. Nevertheless, many editors and other programs still write such a mark at the end of a file, and when such a file gets transferred to another operating system the character suddenly becomes an ordinary source character, namely the character in position 26 of the current font. You can try this, by typing `^^Z` in a document (the double hat is TeX's notation for a control character). In principle, any character may show up in your document due to incorrect transfer protocols between different operating systems, but the 'æ' is probably the most common one. In an earlier *TUGboat* Barbara Beeton discussed the sad story of `^^M` behaving differently on different TeX installations due to operating system differences [BB88].

But I wasn't thinking about file transfer problems. I was talking about macro definitions that all of the sudden produce additional and undesired characters. So here follows my original answer:

The above riddle comes from some real life experience during the implementation of a complex macro for LaTeX3. One evaluation of this macro under `\tracingall` results in more than 700 lines of trace information which made debugging this way somewhat unattractive. Eventually I found the source of these extra characters to be an innocent `\voidb@x` which I had forgotten to remove after changing parts of the code.

For those who never heard of this name, a short explanation: `\voidb@x` is a symbolic name for one of the 256 internal box registers of TeX. It is declared in the `plain.tex` format and, as the name suggests, should always be void, i.e., it's a constant. These symbolic names are declared with a function called `\newbox`⟨*box*⟩ that allocates for ⟨*box*⟩ a new unique box register that later on can be referred to via this name, e.g.,

`\setbox`⟨*box*⟩ `= \hbox{...}`

For other types of registers in TeX there exist similar functions to create symbolic names; for example, `\newcount`⟨*cnt*⟩ makes ⟨*cnt*⟩ a symbolic name referring to some unique internal integer register. But the `\newbox` command is somewhat special. The use of ⟨*box*⟩ declared with it does not mean "use the box register that I represent"; instead it means "typeset the character whose number corresponds to the box register I represent". Only when ⟨*box*⟩ is preceded by a "box" command (like `\setbox`, `\unhbox`, etc.) is it interpreted as a box register. Therefore, a ⟨*box*⟩ out of sequence silently typesets some character in the current font; in such a case it is equivalent to

`\char`⟨*no of the register* ⟨*box*⟩ *represents*⟩

This is a speciality of `\newbox`; all other types of symbolic names are always interpreted as referring to an internal register. For example, a ⟨*dimension*⟩ declared with `\newdimen` which is used out of sequence will be interpreted as an assignment to the dimension register and the following tokens are scanned for a dimension. This will usually result in an error, but if one is unlucky enough (e.g., if a dimension follows) one will get an equally weird behavior.

So far, I have explained this problem as a plain TeX problem, but actually the same might happen to a LaTeX user who declares a "save-box" with `\newsavebox{`⟨*box*⟩`}` and later on uses ⟨*box*⟩ but forgets to call it via the `\usebox` command. The second weird character in the puzzle above was generated this way by saying

```
\newsavebox{\errorbox}
... or ``\errorbox'' in ...
```

From this we can deduce that `\newsavebox` allocated box register 27 for `\errorbox` since this is the font position for "œ" in the Computer Modern fonts.

References

[BB88] Barbara Beeton. Controlling `<ctrl-M>`. *TUGboat* 9(2):182–183 (August 1988).

◇ Frank Mittelbach
Electronic Data Systems (Deutschland) GmbH
Eisenstraße 56 (N15)
D-6090 Rüsselsheim
Federal Republic of Germany
Mittelbach@mzdmza.zdv.Uni-Mainz.de