“A new implementation of the array– and tabular–environments of LaTeX” (TUGboat 9#3) — addenda

Frank Mittelbach
Universität Mainz

1 Corrections to the macros

After submitting the article describing the new implementation of the array– and tabular–environments of LaTeX two errors were found and corrected. The changes below correspond to the items in the figure at the bottom of this page.

- **a.** If a flushright column entry is left empty, a hidden \unskip in the macro \insert@column cancels the stretch (\hfil). The fix is easy. The lines to be altered in the definition of the macro \@classz are below.
- **b.** The second bug was a typo. In the macro \@tfor we should test for \@empty which represents “empty”. The typo \@mpty, however, resolves to “undefined”. Actually this doesn't make any difference. Only the case with the empty argument is no longer optimized. The proper first line in the definition of \@tfor is below.
- **c.** These corrections should also be reflected in the version number.
- **d.** Before the two lines of **c**, we also add a new line of code to avoid reading this file twice.

2 Future versions

After many discussions with LaTeX users I think that it might be better to rename the preamble options t to p (old LaTeX meaning) and p to m (for middle). This will probably change in version 2 coming “sooner or later”[1]. If you are interested in changing the current source at your site you should do so provided you also add a \typeout line which reflects

[1] Quoted from the song “History will teach us nothing” (STING).

a.

```
%    The templates for {\tt l} and {\tt r} (i.e.\ \verb+\@chnum+ $1$ or $2$)
%    are generated the same way. Since one  \verb+\hfil+ is missing
%    the text is moved to the relevant side.
%    \begin{macrocode}
      \d@llar \insert@column \d@llar \hfil \or
      \hfil \d@llar \insert@column \d@llar \or
```

should be changed to

```
%    The templates for {\tt l} and {\tt r} (i.e.\ \verb+\@chnum+ $1$ or $2$)
%    are generated the same way. Since one  \verb+\hfil+ is missing
%    the text is moved to the relevant side.
%    The \verb+\kern\z@+ is needed in case of an empty column
%    entry. Otherwise the \verb+\unskip+ in \verb+\insert@column+
%    removes the \verb+\hfil+.
%    \begin{macrocode}
      \d@llar \insert@column \d@llar \hfil \or
      \hfil\kern\z@ \d@llar \insert@column \d@llar \or
```

b.

```
\def\@tfor#1:=#2\do#3{\def\@fortmp{#2}\ifx\@fortmp\@empty
```

c.

```
\typeout{Style-Option: ‘array’ v1.9k \space\space <30.11.88> (F.M.)}
\typeout{English documentation dated \space\space <30.11.88> (F.M.)}
```

d.

```
\@ifundefined{d@llar}{}{\endinput}
```

the change.[2] You only have to change the two corresponding characters in the `\ifx` tests in the macro `\@testpach`.

The old source (v1.9g) should be available by now at the Heidelberg listserver and in the LaTeX style collection in US listservers.

3 Further developments in Mainz

The following is a short overview concerning the recent activities in Mainz. If there is similar work ongoing at other sites I would be glad to learn about it.

- One year ago I wrote a set of macros extending the theorem–environments of LaTeX which allows the user to use and define `\theoremstyle`'s similar to the `\pagestyle` macro. This code is documented and tested and will be submitted to TUGboat (The bottleneck is the translation to English, sigh!).
- Our main project is to remove the `\protect` macro in LaTeX, because we think this is the most critical feature, especially for novice users. This work is almost finished (only the letter–environment remains to be changed), but the documentation is still rather poor. I hope the code can be released this year ...
- As a reaction to my last article I was asked if I am willing to put the style file used to produce the documentation into the public domain. This is my intention but I feel that contrary to the user interface the inner macros need polishing. Therefore I shall describe the user interface in the next section, to enable others to use this documentation tool in the near future.[3]

4 The user interface of the 'doc' style option

4.1 General conventions

The main characteristic of a TeX file prepared for the 'doc' style option is the use of the `%` character. Every line of documentation starts with this character in column one. Therefore such a (style) file might be used directly by TeX since all comments are bypassed. The documentation style file (`doc.sty`) redefines this character to 'ignore'. In this way documentation comments are made visible to TeX. In this mode we want to format the code 'verbatim' rather than evaluating it. So lines of code are surrounded by

```
%␣␣␣␣\begin{macrocode}
...
%␣␣␣␣\end{macrocode}
```

Note that there must be exactly four spaces between the `%` and the `\end{macrocode}`—TeX is actually looking for this string and not for a macro.

This environment also has a star form which formats the spaces as ␣ just in case somebody needs this.

4.2 Describing a new macro

A macro definition is surrounded by the environment macro which has one argument—the macro name *without* the backslash. This argument is used to generate a main index entry. It is also printed in the left margin prefixed by a backslash. Nesting of this environment is provided (up to three levels) to allow macro definitions within others.

4.3 Index facilities

As mentioned above the argument from every macro environment is used to generate a main index entry. Furthermore all code sections surrounded by the macrocode environment are scanned and every control sequence starting with a backslash will produce a normal index entry for cross referencing purposes[4]. The actual index entries are produced with the macros `\SpecialMainIndex` and `\SpecialIndex`. Both insert a backslash in front of their argument which is also used as the sort key. There also exists a general `\SortIndex` command with two arguments—the sort key and the index entry.

The resulting `idx` file might be sorted by the makeindex program, but other programs can be used if the commands mentioned above are redefined.

4.4 Additional bells and whistles

The verbatim environment is changed to ignore `%` characters. This makes it possible to place examples in the documentation.

There also exist several style parameters which determine the documentation layout.

[2] Other changes are only allowed if the style file is renamed but this is seldom sensible because it increases the number of (normally) unsupported files. See for example the LaTeX style collection. I therefore hope that all improvements, suggestions and/or bug reports are sent back to me.

[3] If someone is interested in using this style file right now, he or she should send me a short message.

[4] This was not implemented in the version used for the last article.