newvbtm and varvbtm
Packages for Variants of `verbatim` Environment*

Hiroshi Nakashima
(Toyohashi Univ. of Tech.)

1999/08/11

Abstract

This file provides two style files; newvbtm to define `verbatim`-like environments; varvbtm to provide set of macros for variants of `verbatim`, e.g. in which `^I` acts as a tab.

Contents

*This file has version number v1.0, last revised 1999/08/11.

1 Introduction

LaTeX users often have trouble when they wish to have their own customized `verbatim`-like environment. Probably you once wished to have an indented-footnotesize-`verbatim` instead of always typing;

```
\begin{itemize}\item[]\footnotesize
\begin{verbatim}
...
\end{verbatim}
\end{itemize}
```

and tried the following just to know it does not work.

```
\newenvironment{myverbatim}{\begin{itemize}\item[]\footnotesize
                              \begin{verbatim}}%
                            {\end{verbatim}\end{itemize}}
```

Another trouble you probablly have had is that what you see in `verbatim` text with `<TAB>` is not what you get because `<TAB>` does not acts as an tab but a space.

Of course it is possible to define your own `verbatim`-like environments if you have enough knowledge of the implementation of `verbatim` including dirty tricks with `\catcode`. However, even a TeXpert should be bored with typing a dirty code like;

```
\begingroup \catcode`\|=0 \catcode`\[=1 \catcode`\]=2
\catcode`\{=12 \catcode`\}=12 \catcode`\\=12
|long|def|@myxverbatim##1\end{myverbatim}[##1|end[myverbatim]]
|endgroup
```

`newvbtm` `varvbtm` The style files distributed with this document will solve these problems. You will have two style files, newvbtm.sty and varvbtm.sty, by processing newvbtm.dtx with docstrip, or simply doing the following.

```
% tex newvbtm.ins
```

The former style provides you `\(re)newverbatim` command to (re)define your own `verbatim`-like environment easily. The latter gives you a set of various macros for tab-emulation, page break control, etc.

2 Usage

2.1 Loading Style Files

Both style files are usable to both LaTeX 2_ε and LaTeX-2.09 users with their standard package loading declaration. If you use LaTeX 2_ε and wish to load, for example, newvbtm, simply do the following.

```
\usepackage{newvbtm}
```

If you still love LaTeX-2.09, the following is what you have to do.

```
\documentstyle[..,newvbtm,...]{⟨main-style⟩}
```

Note that loading varvbtm automatically loads newvbtm too. Thus you may not load both though doing so is safe.

2.2 newvbtm: Define verbatim-like Environments

\newverbatim The command;

```
\newverbatim{⟨env⟩}[⟨n-args⟩]{⟨beg-def-outer⟩}{⟨beg-def-inner⟩}%
                                {⟨end-def-inner⟩}{⟨end-def-outer⟩}
```

defines an environment named ⟨*env*⟩ with ⟨*n-args*⟩ arguments (optionally), and acting conceptually as follows:

⟨*beg-def-outer*⟩`\begin{verbatim}`⟨*beg-def-inner*⟩
⟨*body-of-environment*⟩
⟨*end-def-inner*⟩`\end{verbatim}`⟨*end-def-outer*⟩

Thus to have indented-footnotesize-`verbatim` named, say `indfnsverbatim`, you may simply do the following.

```
\newverbatim{indfnsverbatim}{\begin{itemize}\item[]\footnotesize}{}{}%
                                {\end{itemize}}
```

Since `\newverbatim` defines not only ⟨*env*⟩ but also its starred counterpart ⟨*env*⟩`*` that acts like `verbatim*`, the definition above also defines `indfnsverbatim*` environment.

If you use LaTeX 2ε, you may make ⟨*env*⟩ have an optional argument whose default value is ⟨*default*⟩ by;

```
\newverbatim{⟨env⟩}[⟨n-args⟩][⟨default⟩]{⟨beg-def-outer⟩}{⟨beg-def-inner⟩}%
                                {⟨end-def-inner⟩}{⟨end-def-outer⟩}
```

For example, our `indfnsverbatim` environment can have an optional argument to specify a font size other than `\footnotesize` by the following definition.

```
\newverbatim{indfnsverbatim}[1][\footnotesize]%
        {\begin{itemize}\item[]#1}{}{}{\end{itemize}}
```

The argument ⟨*beg-def-inner*⟩ is for TeXperts who wish to do something overriding what LaTeX's `\verbatim` does. Even if you don't have much confidence in your TeXpertise, however, you can do some useful thing with this argument. For example, the following is obtained by itself.

```
\newverbatim{slverbatim}{}{\slshape}{}{}
```

Also you will find a few commands for this argument in §**??**.

The needs of ⟨*end-def-inner*⟩ is much more limited. One example is to check if `\end{verbatim}` is at the beginning of a line. This examination is done by;

```
\newverbatim{myverbatim}{...}{...}%
        {\ifvmode ⟨at-bol⟩ \else ⟨not-at-bol⟩ \fi}{...}
```

\renewverbatim You may redefine your own `verbatim`-like environment, or even `verbatim` itself, by `\renewverbatim` whose arguments are same as those of `\newenvironment`.

2.3 varvbtm: To Make Variants of `verbatim`

2.3.1 Tab Emulation

`\newtabverbatim` `\renewtabverbatim`

The commands `\(re)newtabverbatim` is to (re)define a `verbatim`-like environment in which `<TAB>` acts as a tab. The syntax of the command is same as that of `\(re)newverbatim`, and its operation is equivalent to;

```
\(re)newverbatim{⟨env⟩}[⟨n-args⟩][⟨default⟩]
        {⟨beg-def-outer⟩}%
        {⟨beg-def-inner⟩⟨beg-def-for-tab⟩}%
        {⟨end-def-for-tab⟩⟨end-def-inner⟩}%
        {⟨end-def-outer⟩}
```

For example;

```
\newtabverbatim{tabverbatim}{}{}{}{}
```

defines `tabverbatim` environment just to make `<TAB>` act as a tab. Another example to have tab emulation version of `indfnsverbatim` with optional argument, say `indfnstabverbatim` is;

```
\newtabverbatim{indfnstabverbatim}[1][\footnotesize]%
        {\begin{itemize}\item[]#1}{}{}{\end{itemize}}
```

Note that in the starred version, e.g. `tabverbatim*`, a `<TAB>` is translated into a sequence of ␣.

`VVBtabwidth`

The distance between tab stops is the width of eight characters of the font used in the environment, i.e. typewriter font usually. If you want to change this default value, set the counter `VVBtabwidth` to the number of characters of the distance.

`\VVBbegintab` `\VVBendtab`

The magical stuff for ⟨*beg-def-for-tab*⟩ and ⟨*end-def-for-tab*⟩ is also accessible through commands `\VVBbegintab` and `\VVBendtab` for TEXperts who wish to do something with `\(re)newverbatim` rather than `\(re)newtabverbatim`.

2.3.2 Form Feed Character

`\VVBprintFF` `\VVBprintFFas`

You might have found that `<FF>` (or `^L`) in `verbatim` caused a mysterious error;

```
! Forbidden control sequence found while scanning use of \@xverbatim.
```

This is because `<FF>` is not *verbatimized*. Giving the command `\VVBprintFF` to ⟨*beg-def-outer*⟩ (or ⟨*beg-def-inner*⟩) of `\newverbatim` does it for you and makes `<FF>` printed as `^L` in default. You may change this default print image by;

```
\VVBprintFFas{⟨str⟩}
```

where ⟨*str*⟩ is a sequence of any printable characters other than `{` and `}`. Note that this command is very *fragile* as `\verb` and `\index`, and thus should not be used in an argument of other commands including `\(re)newverbatim`.

`\VVBbreakatFF` `\VVBbreakatFFonly`

The other way to make `<FF>` acceptable is to give it a useful and natural job, i.e. page breaking. This is done by giving `\VVBbreakatFF` to ⟨*beg-def-inner*⟩ (not *outer*). Its more

powerful relative, `\VVBbreakatFFonly`, is also available to allow page breaking at `<FF>` only. Unfortunately, these two commands are incompatible with `\(re)newtabverbatim` and thus you have to use `\(re)newverbatim` with `\VVBbegintab` *followed by* them.

2.3.3 Non-Verbatim Stuff in `verbatim`-like Environment

`\VVBnonverb` You might have once wished to insert a few non-verbatim stuff, for example math stuff. The command, to be given to ⟨*beg-def-outer*⟩;

```
\VVBnonverb{\⟨char⟩}
```

makes it possible. For example, the author just did the following to produce the result shown above.

```
\newverbatim{verbatimwithnv}{\VVBnonverb{\!}}{}{}{}
\begin{verbatimwithnv}
\VVBnonverb{\!$\langle\mbox{\textit{char}}\rangle$!}
\end{verbatimwithnv}
```

As shown in the example above, the non-verbatim staff is surrounded by a pair of ⟨*char*⟩, the letter '`!`' in this case. Note that ⟨*char*⟩ has to be preceded by '`\`' when it is given as the argument of `\VVBnonbverb`, and ⟨*char*⟩ should not be '`\`'. Also note that the default font for the non-verbatim part is not that for verbatim part, but the font used outside the environment[1].

`\VVBnonverbmath` As mentioned above, math stuffs will be most desirable to be non-verbatim. Thus the macro;

```
\VVBnonverbmath[\⟨char⟩]
```

gives you a shorthand to typeset the stuff surrounded by a pair of ⟨*char*⟩ in math mode. Since the default of ⟨*char*⟩ is `$` as expected, the example above may be;

```
\newverbatim{verbatimwithnv}{\VVBnonverbmath}{}{}{}
\begin{verbatimwithnv}
\VVBnonverb{\$\langle\mbox{\textit{char}}\rangle$}
\end{verbatimwithnv}
```

2.3.4 Verbatim Input

The last thing varvbtm gives you is;

```
\(re)newverbatiminput{⟨command⟩}[⟨n-args⟩][⟨default⟩]%
                     {⟨beg-def-outer⟩}{⟨beg-def-inner⟩}%
                     {⟨end-def-inner⟩}{⟨end-def-outer⟩}
```

to define a ⟨*command*⟩ to `\input` a file. Since this define a ⟨*command*⟩ instead of an environment, ⟨*command*⟩ should have '`\`' as its prefix. The ⟨*command*⟩ has at least one mandatory

[1] Strictly speaking, the font used when `\VVBnonverb` is invoked. Thus if `\VVBnonverb` is preceded by a font changing command, the fond chosen by the command will be used.

argument, ⟨*file*⟩ to be input, which can be referred as first argument if `[`⟨*default*⟩`]` is not supplied, or as second otherwise. Note that, however, if the ⟨*command*⟩ does not have any other arguments, you can omit `[`⟨*n-arg*⟩`]`.

For example;

```
\newverbatiminput{\vinput}{}{}{}{}
```

defines `\vinput{`⟨*file*⟩`}` (and `\vinput*`) that `\input` a ⟨*file*⟩ as if the ⟨*file*⟩ has `\begin`/`\end{verbatim}` at its first and last lines. A little bit more complicated example;

```
\newverbatiminput{\indfnsvinput}[2][\footnotesize]%
        {\begin{itemize}\item[]#1}{}{}{\end{itemize}}
```

defines a indented-footnotesize-by-default version of `\vinput`.

Acknowledgments

The author thanks to Noboru Matsuda and Carlos Puchol whose posts to news groups triggered writing very first version of macros in newvbtm and varvbtm.

For the implementation of these style files, the author refers the base implementations of the macros for `verbatim` environment. These macros are written by Leslie Lamport as a part of LaTeX-2.09 and LaTeX 2ε (1997/12/01) to which Johannes Braams and other authors also contributed.