# newvbtm and varvbtm Packages for Variants of verbatim Environment\*

Hiroshi Nakashima (Toyohashi Univ. of Tech.)

2002/04/08

#### 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 <code>`I</code> acts as a tab.

### Contents

1 Introduction			<b>2</b>	
<b>2</b>	Usa	ige		<b>2</b>
	2.1	Loadir	ng Style Files	2
	2.2	newvb	tm: Define verbatim-like Environments	3
	2.3 varvbtm: To Make Variants of verbatim		m: To Make Variants of verbatim	4
		2.3.1	Tab Emulation	4
		2.3.2	Form Feed Character	4
		2.3.3	Non-Verbatim Stuff in verbatim-like Environment	5
		2.3.4	Verbatim Input	5

<sup>\*</sup>This file has version number v1.1, last revised 2002/04/08.

### 1 Introduction

IATEX 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.

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 T<sub>E</sub>Xpert 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 tabemulation, page break control, etc.

### 2 Usage

### 2.1 Loading Style Files

Both style files are usable to both  $\mbox{LAT}_{E}X 2_{\varepsilon}$  and  $\mbox{LAT}_{E}X-2.09$  users with their standard package loading declaration. If you use  $\mbox{LAT}_{E}X 2_{\varepsilon}$  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.

```
\clinetic{documentstyle[..,newvbtm,...]}{\langle main-style \rangle}
```

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;

 $\label{eq:lasses} $$ \operatorname{l}(env) [(n-args)] {(beg-def-outer)} {(beg-def-inner)} {(end-def-inner)} {(end-def-outer)} }$ 

defines an environment named  $\langle env \rangle$  with  $\langle n-args \rangle$  arguments (optionally), and acting conceptually as follows:

 $\langle beg-def-outer \rangle \setminus begin{verbatim} \langle beg-def-inner \rangle \\ \langle body-of-environment \rangle \\ \langle end-def-inner \rangle \setminus end{verbatim} \langle end-def-outer \rangle$ 

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

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

If you use IAT<sub>E</sub>X  $2_{\varepsilon}$ , you may make  $\langle env \rangle$  have an optional argument whose default value is  $\langle default \rangle$  by;

```
\label{eq:label_linear} $$ \end{tabular} $$ \end{tabula
```

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  $\langle beg-def-inner \rangle$  is for T<sub>E</sub>Xperts who wish to do something overriding what LAT<sub>E</sub>X's \verbatim does. Even if you don't have much confidence in your T<sub>E</sub>Xpertise, 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  $\S 2.3$ .

The needs of  $\langle end\text{-}def\text{-}inner \rangle$  is much more limited. One example is to check if  $\end{verbatim}$  is at the beginning of a line. This examination is done by;

 $\label{eq:linear} $$ \eqref{eq:linear} $$ \eqref{$ 

\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

rerbatim The commands \(re)newtabverbatim is to (re)define a verbatim-like environment rerbatim 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;

For example;

#### $\mbox{newtabverbatim{tabverbatim}{}}$

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  $\Box$ .

VVBtabwidth

\VVBbegintab \VVBendtab 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.

The magical stuff for  $\langle beg-def-for-tab \rangle$  and  $\langle end-def-for-tab \rangle$  is also accessible through commands \VVBbegintab and \VVBendtab for T<sub>E</sub>Xperts 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  $\langle FF \rangle$  is not *verbatimized*. Giving the command  $\langle VVBprintFF$  to  $\langle beg-def-outer \rangle$  (or  $\langle beg-def-inner \rangle$ ) of  $\langle newverbatim$  does it for you and makes  $\langle FF \rangle$  printed as  $^L$  in default. You may change this default print image by;

#### $VVBprintFFas{\langle str \rangle}$

where  $\langle str \rangle$  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  $\langle beg-def-inner \rangle$  (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  $\langle beg-def-outer \rangle$ ;

 $VVBnonverb{\langle char \rangle}$ 

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  $\langle char \rangle$ , the letter '!' in this case. Note that  $\langle char \rangle$  has to be preceded by '\' when it is given as the argument of \VVBnonbverb, and  $\langle char \rangle$  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<sup>1</sup>.

As mentioned above, math stuffs will be most desirable to be non-verbatim. Thus the

\VVBnonverbmath

macro;

 $VVBnonverbmath[\langle char \rangle]$ 

gives you a shorthand to typeset the stuff surrounded by a pair of  $\langle char \rangle$  in math mode. Since the default of  $\langle char \rangle$  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;

to define a  $\langle command \rangle$  to  $\forall input$  a file. Since this define a  $\langle command \rangle$  instead of an environment,  $\langle command \rangle$  should have '\' as its prefix. The  $\langle command \rangle$  has at least one mandatory argument,  $\langle file \rangle$  to be input, which can be referred as first argument if  $[\langle default \rangle]$  is not supplied, or as second otherwise. Note that, however, if the  $\langle command \rangle$  does not have any other arguments, you can omit  $[\langle n-arg \rangle]$ .

For example;

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

defines  $\forall input \{ \langle file \rangle \}$  (and  $\forall input *$ ) that  $input a \langle file \rangle$  as if the  $\langle file \rangle$  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.

<sup>&</sup>lt;sup>1</sup>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.

## 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  $\underline{\text{LATEX}}$ -2.09 and  $\underline{\text{LATEX}} 2_{\varepsilon}$  (1997/12/01) to which Johannes Braams and other authors also contributed.

# Index

Underlined number refers to the page where the specification of corresponding entry is described.

С	S
counters:	style files:
VVBtabwidth $\ldots \ldots \ldots \ldots \ldots $	newvbtm <u>2</u> , 3
D	varvbtm <u>2</u>
D \documentstyle 2	$\mathbf{U}$
(documentstyre	\usepackage $\underline{2}$
$\mathbf{E}$	V
environments:	v
verbatim	varvbtm (style file) $\dots \dots \dots$
	verbatim (environment) 2
Ν	\VVBbegintab <u>4</u>
\newtabverbatim $\ldots \ldots 4$	<b>\VVBbreakatFF</b> $\underline{4}$
	<b>\VVBbreakatFFonly</b> $\underline{4}$
<b>newvbtm</b> (style file) <u>2</u> , 3	\VVBendtab $\underline{4}$
\newverbatim $\ldots \ldots 3$	<b>\VVBnonverb</b>
	VVBnonverbmath
R	\VVBprintFF 4
$\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ $	\VVBprintFFas
\renewverbatim $\dots \dots \dots$	VVBtabwidth (counter) $\ldots \ldots \ldots \underbrace{\overline{4}}_{\underline{4}}$