

# 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 `^I` acts as a tab.

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Usage</b>	<b>2</b>
2.1	Loading Style Files . . . . .	2
2.2	<code>newvbtm</code> : Define <code>verbatim</code> -like Environments . . . . .	3
2.3	<code>varvbtm</code> : To Make Variants of <code>verbatim</code> . . . . .	4
2.3.1	Tab Emulation . . . . .	4
2.3.2	Form Feed Character . . . . .	4
2.3.3	Non-Verbatim Stuff in <code>verbatim</code> -like Environment . . . . .	5
2.3.4	Verbatim Input . . . . .	5

---

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

# 1 Introduction

L<sup>A</sup>T<sub>E</sub>X 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 probably have had is that what you see in `verbatim` text with `<TAB>` is not what you get because `<TAB>` does not act as a 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`      The style files distributed with this document will solve these problems. You will have  
`varvbtm` 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 tabulation, page break control, etc.

## 2 Usage

### 2.1 Loading Style Files

Both style files are usable to both L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> and L<sup>A</sup>T<sub>E</sub>X-2.09 users with their standard package loading declaration. If you use L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> and wish to load, for example, `newvbtm`, simply do the following.

```
\usepackage{newvbtm}
```

If you still love L<sup>A</sup>T<sub>E</sub>X-2.09, the following is what you have to do.

```
\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;

```
\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}{\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  $\text{\LaTeX} 2_{\epsilon}$ , 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}{\footnotesize}\end{itemize}}
```

The argument `<beg-def-inner>` is for  $\text{\TeX}$ pers who wish to do something overriding what  $\text{\LaTeX}$ 's `\verbatim` does. Even if you don't have much confidence in your  $\text{\TeX}$ pertise, however, you can do some useful thing with this argument. For example, the following is obtained by itself.

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

Also you will find a few commands for this argument in §2.3.

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 `TeX`perts 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.



## 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 L<sup>A</sup>T<sub>E</sub>X-2.09 and L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> (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.

<b>C</b>		<b>S</b>	
counters:		style files:	
VVBtabwidth .....	<u>4</u>	newvbtm .....	<u>2</u> , <u>3</u>
		varvbtm .....	<u>2</u>
<b>D</b>		<b>U</b>	
\documentstyle .....	<u>2</u>	\usepackage .....	<u>2</u>
<b>E</b>		<b>V</b>	
environments:		varvbtm (style file) .....	<u>2</u>
verbatim .....	<u>2</u>	verbatim (environment) .....	<u>2</u>
<b>N</b>		\VVBbegintab .....	<u>4</u>
\newtabverbatim .....	<u>4</u>	\VVBbreakatFF .....	<u>4</u>
newvbtm (style file) .....	<u>2</u> , <u>3</u>	\VVBbreakatFFonly .....	<u>4</u>
\newverbatim .....	<u>3</u>	\VVBendtab .....	<u>4</u>
<b>R</b>		\VVBnonverb .....	<u>5</u>
\renewtabverbatim .....	<u>4</u>	\VVBnonverbmath .....	<u>5</u>
\renewverbatim .....	<u>3</u>	\VVBprintFF .....	<u>4</u>
		\VVBprintFFas .....	<u>4</u>
		VVBtabwidth (counter) .....	<u>4</u>