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

Hiroshi Nakashima
(Toyohashi Univ. of Tech.)

翻译:virhuiai
福建师范大学

2002/04/08

摘要

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.

本文件提供了两个样式文件：newvbtm 用于定义类似于 `verbatim` 的环境；varvbtm 提供一组宏，用于处理 `verbatim` 的变体，例如其中 `^I` 作为制表符。

目录

---

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

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
```

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.

介绍

当 LaTeX 用户希望拥有自定义的类似于 `verbatim` 的环境时，通常会遇到困难。也许您曾经希望有一个缩进的脚注大小的 `verbatim`，而不是每次都要输入以下内容：

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

并尝试了以下代码，只是发现它无法正常工作：

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

您可能遇到的另一个问题是，使用 `<TAB>` 在 `verbatim` 文本中所看到的不是所获得的，因为 `<TAB>` 并不像制表符一样起作用，而是一个空格。

当然，如果您对包括使用 `\catcode` 进行一些“脏技巧”在内的 `verbatim` 的实现有足够的了解，那么您可以自定义自己的 `verbatim` 环境。然而，即使是 TeX 专家，也会对输入以下这种“脏代码”感到厌烦：

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

本文档附带的样式文件将解决这些问题。通过处理 newvbtm.dtx 并使用 docstrip，您可以得到两个样式文件：newvbtm.sty 和 varvbtm.sty，或者只需要执行以下命令：

`newvbtm`
`varvbtm`

```
% tex newvbtm.ins
```

前者提供了 `\(re)newverbatim` 命令，可以轻松（重新）定义自己的 `verbatim` 类似环境。后者为您提供了一组用于模拟制表符、控制分页等的各种宏。

2 Usage

2.1 Loading Style Files

Both style files are usable to both LaTeX 2$_\varepsilon$ and LaTeX-2.09 users with their standard package loading declaration. If you use LaTeX 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.

```
\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

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 2$_\varepsilon$, you may make ⟨*env*⟩ have an optional argument whose default value is ⟨*default*⟩ by;

使用方法

2.1 加载样式文件

这两个样式文件都适用于LaTeX 2$_\varepsilon$和 LaTeX-2.09 用户，并且可以使用它们的标准包加载声明。如果你使用LaTeX 2$_\varepsilon$并希望加载，例如 newvbtm，只需执行以下操作。

```
\usepackage{newvbtm}
```

如果你仍然在使用 LaTeX-2.09，你需要执行以下操作。

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

请注意，加载 varvbtm 会自动加载 newvbtm。因此，尽管可以安全地这样做，但不要同时加载两者。

2.2 newvbtm：定义类似 `verbatim` 的环境

命令 `\newverbatim`

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

定义了一个名为⟨*env*⟩的环境，有⟨*n-args*⟩个参数（可选），并且在概念上的作用如下:

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

因此，要定义一个名为 `indfnsverbatim` 的缩进脚注大小的 `verbatim` 环境，只需执行以下操作。

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

由于`\newverbatim` 不仅定义了⟨*env*⟩,还定义了和`verbatim*`类似的星号版本⟨*env*⟩`*`,上述定义也定义了`indfnsverbatim*` 环境。

如果你使用LaTeX 2$_\varepsilon$,你可以通过以下方式使⟨*env*⟩具有可选参数,其默认值为⟨*default*⟩。

```
\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 §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}{...}
```

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

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⟩}
```

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

例如,我们的`indfnsverbatim`环境可以通过以下定义具有可选参数,以指定除`\footnotesize`之外的字体大小。

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

⟨*beg-def-inner*⟩参数是给希望覆盖 LaTeX 的`\verbatim` 命令的 TeX 专家使用的。然而，即使你对自己的 TeX 专业知识没有太多信心，你也可以使用这个参数做一些有用的事情。例如，以下内容是通过以下命令得到的。

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

此外，你还可以在第 2.3节中找到一些用于此参数的命令。

⟨*end-def-inner*⟩参数的需求要少得多。一个例子是检查`\end{verbatim}` 是否位于一行的开头。可以通过以下方式进行检查:

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

`\renewverbatim`

2.3 **varvbtm:** 生成 `verbatim` 的变体

2.3.1 模拟制表符

命令`\(re)newtabverbatim` 用于（重新）定义一个类似于`verbatim` 的环境，其中`<TAB>` 被视为制表符。该命令的语法与`\(re)newverbatim` 相同，其操作与其等效； `\newtabverbatim` `\renewtabverbatim`

```
\(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 ␣.

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 ⟨*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

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`.

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.

例如

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

定义了`tabverbatim` 环境，使得`<TAB>` 被视为制表符。还可以通过以下示例定义具有可选参数的`indfnsverbatim` 的模拟制表符版本，例如`indfnstabverbatim`；

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

请注意，在星号版本中，例如`tabverbatim*`，`<TAB>` 被转换为一个空格序列␣。

制表符之间的距离是环境中使用的字体的八个字符的宽度，即通常的等宽字体。如果要更改此默认值，请将计数器`VVBtabwidth` 设置为距离的字符数。

`VVBtabwidth`

对于希望对`\(re)newverbatim` 进行操作而不是对`\(re)newtabverbatim` 进行操作的 TEX 专家，可以通过命令 `\VVBbegintab` 和 `\VVBendtab` 访问 ⟨*beg-def-for-tab*⟩ 和 ⟨*end-def-for-tab*⟩ 的神奇内容。

`\VVBbegintab`
`\VVBendtab`

2.3.2 换页符

您可能发现在`verbatim` 中使用的`<FF>`（或`^L`）会导致一个神秘的错误；

`\VVBprintFF`
`\VVBprintFFas`

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

这是因为`<FF>` 没有被 *verbatimized*。给出`\VVBprintFF` 命令到 ⟨*beg-def-outer*⟩（或⟨*beg-def-inner*⟩）中的`\newverbatim` 可以为您完成这个操作，并且默认情况下将`<FF>` 打印为`^{L}`。您可以通过以下方式更改此默认打印图像；

```
\VVBprintFFas{⟨str⟩}
```

其中⟨*str*⟩是除了`{`和`}` 之外的任何可打印字符的序列。请注意，此命令非常脆弱，就像`\verb` 和`\index` 一样，因此不应在其他命令的参数中使用，包括`\(re)newverbatim`。

另一种使 `<FF>` 可接受的方法是给它一个有用和自然的任务，即分页。这是通过将 `\VVBbreakatFF` 给予 ⟨*beg-def-inner*⟩（而不是 *outer*）来实现的。它的更强大的形式 `\VVBbreakatFFonly` 也可用于仅在 `<FF>` 处分页。不幸的是，这两个命令与 `(re)newtabverbatim` 不兼容，因此您必须使用 `(re)newverbatim` 与 `\VVBbegintab` *followed by* 它们。

`\VVBbreakatFF`
`\VVBbreakatFFonly`

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

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 but the font used outside the environment[1].

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}
```

---

[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.

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⟩}
```

2.3.3 非抄录环境中的非抄录内容

你可能曾经希望插入一些非抄录内容,比如数学内容。可以通过给予⟨*beg-def-outer*⟩的命令来实现;

`\VVBnonverb`

```
\VVBnonverb{\⟨char⟩}
```

这样就可以了。例如，作者就是通过以下方式生成了上面显示的结果。

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

如上面的示例所示,非抄录部分被一对⟨*char*⟩包围,这里是'!' 字符。注意,当⟨*char*⟩被作为`\VVBnonbverb` 的参数给出时，应该在其前面加上 ',´ 而且⟨*char*⟩不能是'。´ 另外，请注意非抄录部分的默认字体不是抄录部分的字体，而是环境外部使用的字体。

如上所述，非抄录的数学内容是最理想的。因此，可以使用以下宏；

`\VVBnonverbmath`

```
\VVBnonverbmath[\⟨char⟩]
```

它让你可以很方便地在数学模式下生成一对⟨*char*⟩包围的内容。由于⟨*char*⟩的默认值是期望的 `$`，所以上面的示例也可以写成：

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

2.3.4 抄录输入

最后一件事情是 varvbtm 给你提供的：

```
\(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 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 2$_\varepsilon$ (1997/12/01) to which Johannes Braams and other authors also contributed.

这样可以定义一个⟨*command*⟩来用于`\input` 文件。由于定义的是⟨*command*⟩而不是环境，所以⟨*command*⟩应该以’ 作′ 为前缀。至少有一个必需的参数⟨*file*⟩要被输入，如果没有提供 `[`⟨*default*⟩`]`，则可以将其作为第一个参数引用，否则作为第二个参数引用。然而，请注意，如果⟨*command*⟩没有其他参数，可以省略 `[`⟨*n-arg*⟩`]`。

例如：

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

定义了`\vinput{`⟨*file*⟩`}`（以及`\vinput*`），它们以`\begin`/`\end{verbatim}` 作为⟨*file*⟩的第一行和最后一行进行`\input`。再举一个略微复杂的例子：

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

定义了一个默认为缩进的 `\vinput` 的 `footnotesize` 版本。

致谢

作者感谢 Noboru Matsuda 和 Carlos Puchol，他们在新闻组中的帖子触发了在 newvbtm 和 varvbtm 中编写宏的第一个版本。

对于这些样式文件的实现，作者参考了 `verbatim` 环境的宏的基本实现。这些宏是由 Leslie Lamport 编写的，作为 LaTeX-2.09 和 LaTeX 2$_\varepsilon$（1997/12/01）的一部分，Johannes Braams 和其他作者也做出了贡献。