The verbatimcopy package*

Lars Madsen† Ulrich Diez‡

November 17, 2008

Intro

The verbatimcopy package is a package that enables LaTeX to make verbatim copies of files. Additionally one can specify which (sub)directory to save the copy in.

Please note that this version is not backwards compatible! As `\VerbatimCopy` ***now can handle a*** `\verb`***-like syntax, it can no longer be used inside other macros. Though we do provide a compatability mode that reverts back to the old functionality. See the* Compatability *section for more information.***

Usage

The package provides two macros. Firstly the

`\setOutputDir` `\setOutputDir{`⟨*dir*⟩`}`

macro is used for specifying which default directory to copy files to. `\setOutputDir` will automatically add a trailing »/« if missing. On systems that are not able to use »/« as a directory delimiter, please redefine `\VC@slash` accordingly.[1] The default output directory is the current directory. Please note that we cannot create directories, the directory specified has to exist beforehand.

The main macro of the package is

`\VerbatimCopy` `\VerbatimCopy{`⟨*input file*⟩`}{`⟨*output file*⟩`}`

which will take a verbatim copy of ⟨*input file*⟩ and save it as ⟨*output file*⟩. *Note:* If ⟨*output file*⟩ does not have an extension, then LaTeX gives it »`.tex`«.

Be very careful when using this package, it will overwrite the target file without warning.

*Version 0.06

†Email: daleif@imf.au.dk (packing and user docs)

‡Email: eu_angelion@web.de (macro coding)

[1] As we do not have access to systems where this is needed this features has not been throughly tested.

Advanced usage

Actually both `\setOutputDir` and `\VerbatimCopy` supports a more `\verb`-like syntax. That is you can use a syntax similar to

`\setOutputDir`⟨*char*⟩⟨*some name not containing char*⟩⟨*char*⟩

example

```
\setOutputDir|# #{@} b&~ $$/|
```

for the already existing dir »`# #{@} b&~ $$`«, meaning that we can use directories with special names. Any linebreak in the directory name or in the input or output names for `\VerbatimCopy` are silently ignored, but the spaces do count. Many editors automatically strip spaces at line-endings, thus avoiding linebreaks after spaces might be a good idea.

With `\VerbatimCopy` you will need two sets of ⟨*char*⟩ characters (to replace the braces in the syntax above) and these sets need not be the same. Example:

```
\VerbatimCopy#test.tex#+test%#2}.tex+
```

meaning that the file »`test%#2}.tex`« will be a verbatim copy of the original file »`test.tex`«.

A note about spaces in filenames

Heiko Oberdiek mentions in his `grffile` package:

> *In general it is not possible to use space inside file names, because TEX considers the space character as termination in its syntax for commands that expect a file name.*

But in most modern LaTeX distributions one can use double quotes around the filename containing spaces. If you use

```
\usepackage[enquotefilenames]{verbatimcopy}
```

file-paths will get wrapped into double-quotes automatically and you can avoid doing this “by hand” for the sake of avoiding issues which might come from string-concatenation of `\setOutputDir`'s ⟨*dir*⟩ and `\VerbatimCopy`'s ⟨*output file*⟩.

On systems that allow/require a different file-name-delimiter, redefine `\VC@quote` accordingly.

Compatability

As mentioned ealier the macros `\VerbatimCopy` and `\setOutputDir` have been changed to allow a `\verb`-like functionality. This also means that `\VerbatimCopy` and `\setOutputDir` can no longer be used inside other macros. Some people might be able to live without the `\verb`-like features and would like to be able to use `\VerbatimCopy` and `\setOutputDir` inside other macros. These users have two possibilities:

```
\usepackage[compatibility]{verbatimcopy}
```

or

```
\OldVerbatimCopy
```

```
\OldsetOutputDir
```

(which is always available). Just note that using the 'old' macros you cannot have special characters in the filenames, but spaces are still allowed using the syntax shown ealier (using double quotes).

A hint for programmers

As mentioned earlier `\VerbatimCopy` can no longer be used, as is, inside other macros. For that you will need a trick. Here is what Ulrich Diez provided such that this version of `verbatimcopy` works with my LaTeX-book:

```
\makeatletter
% helper macro
\newcommand\VCverbdef[1]{%
    \VCverbaction{\expandafter\newcommand
                  \expandafter*\expandafter
                  #1\@firstofone}{}%
}%
%
% use this incase you use odd input or output filenames
\VCverbdef\felineFC@input|input with odd letter.tex|
\VCverbdef\felineFC@output+output+
% I use 'feline' for localisation in some internal macros
\newcommand\feline@run@verbatimcopy{%
  \@bsphack
  \begingroup
  \edef\@tempa{%
    \endgroup
    \noexpand\scantokens{%
      \begingroup
      \noexpand\csname VerbatimCopy\endcsname
      "\felineFC@input""\felineFC@output"%
      %\scantokens will produce a trailing space without this:
      \endlinechar=-1
    }%
    \endgroup
    \@esphack
  }\@tempa
}
```

Acknowledgement

The main idea behind this package was originally posted on `comp.text.tex` by Ulrich Diez (`http://groups.google.com/group/comp.text.tex/msg/153e3dc7a8d0e548?hl=en`). The idea is to use a simple, local, redefinition of `\verbatiminput`, which, instead of writing the contents in the document, just writes it to another file.

Later Ulrich Diez provided new code that extends the original to macros. This is the code that are currently in the `verbatimcopy` package, completely replacing the old code.

The verbatimcopy package automatically loads the `verbatim` package.

Why?

One might ask why the need for a functionality like this?

Here is where I use it: In my Danish 400+ pages LaTeX book (`http://www.imf.au.dk/system/latex/bog`), I have lots of examples. For many of these examples I'd like to offer the source for download (as was done for the LaTeX Companion, 2nd edition). Several examples are created using autogenerated fully compilable files that are converted into EPS/PDF and included as graphics. All of my examples are of course numbered, but the actual files that make up the external files are all individually named. To make it easier for users to find the right code, the downloadable source files are named to fit the example numbers in the book. So instead of writing a script to rename files, we simply let LaTeX do the verbatim copying from one directory into another.

(TODO: fingure out a way to see if I actually need to copy a file or write it to disk, the XY-Pic package can do something similar.)