The \texttt{cmdstring} Package

Josselin Noirel

23rd April 2007 (v1.1)

\textbf{Abstract}

For programmers, it is quite common to use the name of a macro (for instance ‘foo’ for the macro $\texttt{\textbackslash foo}$, in other words, the letters $f$, $o$, and $o$). For this purpose, it is customary to call $\texttt{\string}$ which does this sort of things . . . except that it prepends the escape character, controlled by the counter $\texttt{\escapechar}$, in front of the name. The traditional ways of getting rid of it, is to set $\texttt{\escapechar}$ to $-1$ (without forgetting to restore the proper value at the end of the process) or to make use of $\texttt{\@gobble}$. The former has the drawback not to be expandible while the second isn’t completely reliable. Here we extend the expandible methods to make it reliable.

Use this package with

\begin{verbatim}
\usepackage{cmdstring}
\end{verbatim}

or

\begin{verbatim}
\RequirePackage{cmdstring}
\end{verbatim}

Then whatever the value of $\texttt{\escapechar}$, $\texttt{cmdstring \langle command \rangle}$ will give correct result. For instance ‘$\texttt{cmdstring \documentclass}$’ gives ‘documentclass’; ‘$\texttt{cmdstring \textbackslash}$’ gives ‘\’, ‘$\texttt{cmdstring \textbackslash \textbackslash}$’ gives ‘\’, i.e., the command name without the escape character.
A small test can be performed for all representative values of $\backslash$escapechar

\count255 = -11
\loop \backslash$escapechar = \count255
  \[\backslash$cmdstring \\]
  \[\backslash$cmdstring \ \]
  \[\backslash$cmdstring \ \]
  \ifnum \count255 < 300 \\advance \count255 by 1
  \repeat

which gives