The Macroswap Package

Robert J Lee
latex@rjlee.homelinux.org
August 21, 2013

Abstract
This package provides macros to allow the user to redefine a pair of macros so that the token lists they expand to will be swapped.

1 Introduction
This package was written to allow a simple, readable syntax for swapping the values within macros, used as program variables.

Usage
\macroswap
The \macroswap macro allows you to temporarily swap two macro definitions; for example:
\newcommand{\myfirst}{1}
\newcommand{\myend}{2}
\begingroup
macroswap{myfirst}{myend}
\myfirst\myend % prints ‘‘21’’
\endgroup
\myfirst\myend % prints ‘‘12’’ (swapped definitions were local)

This local swap is often unwanted, and can use up TeX’s stack memory. So you may prefer the global version:
\gmacroswap
The \gmacroswap macro allows you to globally swap two macro definitions; for example:
\newcommand{\myfirst}{1}
\newcommand{\myend}{2}

*This document corresponds to Macroswap ?, dated ?.
Usage with arrayjobx

To globally swap two elements of an array declared with the arrayjobx package, you need to arrange to swap the macros used by that package. This is not part of the public interface and may change at any time; unfortunately, it does not seem possible to do this with only the public interface, especially if you do not want to expand the tokens in the array.

Assuming arrayjobx version 1.04 (labelled “05/03/2010”), it is only necessary to swap the values of the underlying macros using the following syntax:

\newarray{arr}
\readarray{arr}{A&C&B}
\gmacroswap{arr2\string~}{arr3\string~}
\arr(1)\par % prints ‘‘A’’
\arr(2)\par % prints ‘‘B’’
\arr(3)\par % prints ‘‘C’’

Implementation

\gmacroswap

Given the name of two macros, swap their definitions. Produces no output. This version uses plain \let to produce a local assignment.

1 \newcommand{\macroswap}[2]{%
NB: there’s no guarantee that \#1 or \#2 will be a single token.
First verify that they expand to a single macro name:
2 \expandafter ifcsname#1endcsname%
3 \else%
4 \PackageError{macrocode}{Two macro names are required for macroswap %
5 (got ‘‘#1’’ and ‘‘#2’’)}{%
6 Two macro names are required, but the first parameter does not %
7 expand to a defined macro name. Macros have not been swapped.}%
8 \fi%
9 \expandafter ifcsname#2endcsname%
10 \else%
11 \PackageError{macrocode}{Two macro names are required for macroswap %
12 (got ‘‘#1’’ and ‘‘#2’’)}{%
13 Two macro names are required, but the second parameter does not %
14 expand to a defined macro name. Macros have not been swapped.}%
15 \fi%
First we use a temporary variable, \texttt{macroswap@temp}, to hold the value of the first macro.

\texttt{\expandafter} is used to evaluate the \texttt{\csname...\endcsname} become the \texttt{\let}, as \texttt{\let} will only redefine its first parameter.

16 \texttt{\expandafter\let\expandafter\macroswap@temp\csname#1\endcsname}\%  
Next, the same method is used to reassign each passed-in macro.

17 \texttt{\expandafter\let\csname#1\expandafter\endcsname\csname#2\endcsname}\%  
18 \texttt{\expandafter\let\csname#2\endcsname\macroswap@temp}\%

Finally, we throw away our temporary macro:

19 \texttt{\let\macroswap@temp\relax}\%
20 \}

\texttt{\macroswap}  
Given the name of two macros, swap their definitions. Produces no output.  
This version is global; the only difference here is that \texttt{\global} is used to modify \texttt{\let} to applied globally.

21 \texttt{\newcommand{\gmacroswap}[2]{\%}\}
22 \texttt{\expandafter\let\expandafter\macroswap@temp\csname#1\endcsname}\%
NB: \texttt{\expandafter} is used before \texttt{\global} so that the \texttt{\let} becomes global...

23 \texttt{\expandafter\global\expandafter\let\csname#1\expandafter\endcsname\csname#2\endcsname}\%  
24 \texttt{\expandafter\global\expandafter\let\csname#2\endcsname\macroswap@temp}\%  
25 \texttt{\let\macroswap@temp\relax}\%
26 \}

That is all

\textbf{Change History}

\texttt{v1.0}
0.0 General: Initial version \ldots \ldots 1

\texttt{v1.1}
0.1 General: Correct month in version date, remove (unstated) dependency on \texttt{etoolbox} package, support swapping macros that expect parameters, improved error reporting (thanks to Christian Tellechea, Ulrich Diez and Dan) \ldots \ldots 1
Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

\else \hspace{.2em} 3, 10
\global \hspace{.2em} 23, 24 \macroswap \hspace{.2em} 1, 1, 21
\expandafter \hspace{.2em} 2, 9, 16–18, 22–24
\gmacroswap \hspace{.2em} 1, 1, 21 \macroswap@temp 16, 18, 19, 22, 24, 25
\fi \hspace{.2em} 8, 15 \ifcsname \hspace{.2em} 2, 9 \PackageError \hspace{.2em} 4, 11