

# 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 defininions 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 ?.

```

\begingroup
\macroswap{myfirst}{myend}
\myfirst\myend % prints ‘21’
\endgroup
\myfirst\myend % prints ‘21’ (swapped defininions are global)

```

## 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}
\macroswap{arr2\string~}{arr3\string~}
\arr(1)\par % prints ‘A’
\arr(2)\par % prints ‘B’
\arr(3)\par % prints ‘C’

```

## Implementation

`\macroswap` 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#1\endcsname%
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#2\endcsname%
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, `macroswap@temp`, to hold the value of the first macro.

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

```
16 \expandafter\let\expandafter\macroswap@temp\csname#1\endcsname%
```

Next, the same method is used to reassign each passed-in macro.

```
17 \expandafter\let\csname#1\expandafter\endcsname\csname#2\endcsname%
```

```
18 \expandafter\let\csname#2\endcsname\macroswap@temp%
```

Finally, we throw away our temporary macro:

```
19 \let\macroswap@temp\relax%
```

```
20 }
```

`\macroswap` Given the name of two macros, swap their definitions. Produces no output.

This version is global; the only difference here is that `\global` is used to modify `\let` to applied globally.

```
21 \newcommand{\gmacroswap}[2]{%
```

```
22 \expandafter\let\expandafter\macroswap@temp\csname#1\endcsname%
```

NB: `\expandafter` is used before `\global` so that the `let` becomes global...

```
23 \expandafter\global\expandafter\let\csname#1\expandafter\endcsname\csname#2\endcsname%
```

```
24 \expandafter\global\expandafter\let\csname#2\endcsname\macroswap@temp%
```

```
25 \let\macroswap@temp\relax%
```

```
26 }
```

That is all

## Change History

v1.0

General: Initial version . . . . . 1

v1.1

General: Correct month in version date, remove (unstated) dependency on `etoolbox` pack-

age, support swapping macros that expect parameters, improved error reporting (thanks to Christian Tellechea, Ulrich Diez and Dan) . . . . . 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.

<b>E</b>		<b>G</b>		<b>M</b>	
<code>\else</code> . . . . .	3, 10	<code>\global</code> . . . . .	23, 24	<code>\macroswap</code> . . . .	1, <i>1</i> , <u>21</u>
<code>\expandafter</code> . . . . .		<code>\gmacroswap</code> . . . .	<u>1</u> , <i>1</i> , 21	<code>\macroswap@temp</code>	16,
	2, 9, 16–18, 22–24				18, 19, 22, 24, 25
<b>F</b>		<b>I</b>		<b>P</b>	
<code>\fi</code> . . . . .	8, 15	<code>\ifcsname</code> . . . . .	2, 9	<code>\PackageError</code> . . . .	4, 11