The `suffix` Package
Version 1.5a

David Kastrup[*]

2006/07/15

1 Basics

The `suffix` package has the purpose of making it easy to define and maintain command variants like `\macro*` and even `\macro\/` or similar. It requires eTEX version 2 for its work. The suffixes are fetched with `\futurelet`, so things like `\bgroup` and `{` can't be distinguished. In addition, the efficiency depends on the complexity of the suffix' definition, so you should preferably use characters or short commands as a suffix. A suffixed command itself counts as short for this purpose. How does a suffix definition look like?

`\WithSuffix` The general form is

`\WithSuffix`⟨*prefixed definition*⟩⟨*macro*⟩⟨*suffix*⟩...

where ⟨*prefixed definition*⟩ is something like `\xdef`, `\global\let` or similar. Recognised prefixes are `\global`, `\long`, `\protected` (the latter is rarely useful, as the original definition already is robust), and `\expandafter` (with its 'natural' meaning), specially recognized commands are `\gdef` and `\xdef`. Other commands can be used as long as they are suitable as an undelimited macro argument. This means they must either be a single token like `\newcommand` or brace-enclosed like `{\newcommand*}`. ⟨*macro*⟩ can be a macro or an active character. It should be a single token suitable for assignment with `\let`. ⟨*suffix*⟩ can be something like a single letter such as `*` or `[`.

For example, assume that a command `\snarf` already exists and we want to define a variant `\snarf[`⟨*optarg*⟩`]`. Then we can do this with

`\WithSuffix\long\def\snarf[#1]{`⟨*Definition using #1*⟩`}`

That's pretty much it. Oh, only when a command is recognised as having a prefix `\global` or being `\xdef` or `gdef` will the corresponding redefinitions be done globally. This is rarely a concern.

`\SuffixName` In case you need to refer to the control sequence name used to refer to the suffixed macro, you can access it as

[*] Email: `dak@gnu.org`

```
\csname\SuffixName⟨macro⟩⟨suffix⟩\endcsname
```

\NoSuffixName and if you need to refer to the original unsuffixed macro, you can access it as

```
\csname\NoSuffixName⟨macro⟩\endcsname
```

2 The driver file for the documentation

Installation is done by `bigfoot.ins`, so look there for more information for that. Here comes the documentation driver.

```
⟨*driver⟩
\documentclass{ltxdoc}
\usepackage{hyperref}
\usepackage{suffix}
\begin{document}
\OnlyDescription
⟨driver⟩ \AlsoImplementation
\DocInput{suffix.dtx}
\end{document}
⟨/driver⟩
```

3 Implementation

First we announce the package and check for eTEX 2.

```
⟨*style⟩
\ProvidesPackage{suffix}[2006/07/15 1.5a Variant command support]
\ifcase\ifx\eTeXversion\@undefined \@ne\fi
  \ifnum\eTeXversion<\tw@ \@ne\fi\z@
\else
  \PackageError{suffix}{This package requires eTeX version 2}%
  {You might try to use the ‘elatex’ command.}%
\fi
```

\WithSuffix Then we define the `\WithSuffix` command. We use `\@temptokena` to collect prefixes and let `\WSF@global` to `\global` for global definitions.

```
\def\WithSuffix{\@temptokena{}\let\WSF@global\relax
  \WSF@sfx}
```

\WSF@sfx
\WSF@append
\WSF@gobblenext

After checking all prefixes and stuff (we'll fill in this missing link later), we add the defining command itself to the token list, place ⟨*macro*⟩ into `\reserved@a` and fetch ⟨*suffix*⟩ into `\reserved@b`.

```
\long\def\WSF@sfx#1#2{\WSF@append{#1}\def\reserved@a{#2}%
  \afterassignment\WSF@decsuff \WSF@gobblenext}

\def\WSF@append#1{\@temptokena\expandafter{\the\@temptokena#1}}

\def\WSF@gobblenext{\let\reserved@b= }
```

\SuffixName
\NoSuffixName

While we are at it, let us define the macro names to use for suffixed and unsuffixed ⟨*macro*⟩.

```
\long\def\SuffixName#1{WSF:\string#1 \meaning}
\def\NoSuffixName{WSF:\string}
```

\WSF@decsuff

We first check whether the macro has already been suffixed. If it hasn't, we save a copy of it and redefine it in terms of `\WSF@suffixcheck`.

```
\def\WSF@decsuff{\ifcsname
    \expandafter\NoSuffixName\reserved@a\endcsname
  \else
    \WSF@global\expandafter\let\csname
       \expandafter\NoSuffixName\reserved@a
      \expandafter\endcsname \reserved@a
    \long\def\reserved@c##1{\WSF@global\protected\def
      ##1{\WSF@suffixcheck##1}}%
    \expandafter\reserved@c\reserved@a
  \fi
```

Once we have done that, we are ready for calling the definition command on the suffixed ⟨*macro*⟩.

```
  \WSF@global
    \the\expandafter\@temptokena\csname
    \expandafter \SuffixName
    \reserved@a\reserved@b\endcsname}
```

\WSF@suffixcheck

We now do the runtime code. This is executed in a group of its own in order not to interfere with any other macros.

```
\def\WSF@suffixcheck#1{\begingroup\def\reserved@a{#1}%
  \futurelet\reserved@b\WSF@suffixcheckii}
```

\WSF@suffixcheckii

After assigning the ⟨*suffix*⟩ to `\reserved@b`, we split into the case of known and unknown suffix. We don't code this inline, since `\reserved@` in a false conditional might confuse TEX if it happens to be something like `\iffalse` itself.

```
\def\WSF@suffixcheckii{\ifcsname \expandafter\SuffixName
    \reserved@a\reserved@b\endcsname
       \expandafter
       \WSF@suffixcheckiii
    \else
      \expandafter
      \WSF@suffixcheckiv
    \fi}
```

\WSF@suffixcheckiii
\WSF@suffixcheckiv

Calling the macros is reasonably straightforward, we just have to take care not to close the group at the wrong time.

```
\def\WSF@suffixcheckiii{%
  \afterassignment\endgroup
  \expandafter\aftergroup
    \csname \expandafter \SuffixName\reserved@a\reserved@b\endcsname
    \WSF@gobblenext}
```

```

\def\WSF@suffixcheckiv{%
    \expandafter\endgroup
    \csname \expandafter\NoSuffixName\reserved@a\endcsname}
```

\WSF@sfx Now we just augment \SF@sfx to recognize all prefixes and global commands:

```
\WithSuffix\def\WSF@sfx\long{\WSF@append\long\WSF@sfx}
\WithSuffix\def\WSF@sfx\global{\let\WSF@global\global\WSF@sfx}
\WithSuffix\def\WSF@sfx\protected{\WSF@append\protected\WSF@sfx}
\WithSuffix\def\WSF@sfx\expandafter{\expandafter\WSF@sfx\expandafter}
\WithSuffix\edef\WSF@sfx\gdef{\let\WSF@global\global
  \expandafter\noexpand\csname\NoSuffixName\WSF@sfx\endcsname\def}
\WithSuffix\edef\WSF@sfx\xdef{\let\WSF@global\global
  \expandafter\noexpand\csname\NoSuffixName\WSF@sfx\endcsname\edef}
⟨/style⟩
```