The gobble Package

Martin Scharrer
martin@scharrer-scharrer.de

CTAN: http://www.ctan.org/pkg/gobble

VC: https://bitbucket.org/martin_scharrer/gobble

Version v0.2 – 2019/01/04

Abstract

The `gobble` LaTeX package provides more macros to discard (gobble) macro arguments. These macros are by default only for package and class writers, but are exported to the user level by the `gobble-user` package. The macros are also available as generic packages for other TeX formats.

1 Macros

1.1 Macros for Package/Class Writers

The following macros are provided by the generic `gobble.tex` and the LaTeX package `gobble`. Some are already provided by the LaTeX kernel.

```
\@gobble
\@gobbletwo
\@gobblethree
\@gobblefour
```

Gobbles one, two, three or four mandatory arguments.

```
\@gobbleopt
\@gobbletwoopt
\@gobbleallopt
```

Gobbles one, two or all found optional arguments if present.

```
\@gobbletwoopttwo
```

Gobbles (up to) two optional arguments if present and then two mandatory arguments.

`\@firstofone`

Reads one argument and expands to it, i.e. removes the braces around it.

`\@firstoftwo`

Reads two arguments and expands to the first one while discarding the second one.

`\@secondoftwo`

Reads two arguments and expands to the second one while discarding the first one.

`\@firstofthree`
`\@secondofthree`
`\@thirdofthree`

These macros read three arguments and expand to the first, second or third one, respectively, while discarding the other.

`\@gobbletofi{`⟨*code*⟩`}`⟨*discarded code*⟩`\fi`
`\@gobbletoelse{`⟨*code*⟩`}`⟨*discarded code*⟩`\else`
`\@gobbletoor{`⟨*code*⟩`}`⟨*discarded code*⟩`\or`

These macros read one argument and then gobble everything to the next `\fi`, `\else` or `\or`, respectively. Afterwards they expand to the first argument. This allows to break out of an `\if..` or `\ifcase` branch, which is sometimes required if code should be executed after the conditional.

1.2 User Level Macros

The following macros are provided by the generic `gobble-user.tex` and the LaTeX package `gobble-user`. These also load the `gobble.tex` file or package `gobble` package, respectively.

They macros are identical to the corresponding versions with the leading '`@`', but can be used freely in a user document.

`\gobble`
`\gobbletwo`
`\gobblethree`
`\gobblefour`

`\gobbleopt`
`\gobbletwoopt`
`\gobbleallopt`

```
\gobbletwoopttwo
```

```
\firstofone
```

```
\firstoftwo
\secondoftwo
```

```
\firstofthree
\secondofthree
\thirdofthree
```

```
\gobbletofi
\gobbletoelse
\gobbletoor
```

2 Implementation

```
%<!COPYRIGHT>
\NeedsTeXFormat{LaTeX2e}[1999/12/01]
\ProvidesPackage{gobble}[%
%<!DATE>
%<!VERSION>
%<*DRIVER>
    2099/01/01 develop
%</DRIVER>
    Provides more gobble macros]

\input{gobble}

%<!COPYRIGHT>
\NeedsTeXFormat{LaTeX2e}[1999/12/01]
\ProvidesPackage{gobble-user}[%
%<!DATE>
%<!VERSION>
%<*DRIVER>
    2099/01/01 develop
%</DRIVER>
    Provides user level macros]

\RequirePackage{gobble}
\input{gobble-user}

\expandafter\ifx\csname gobble.tex loaded\endcsname\relax
    \expandafter\def\csname gobble.tex loaded\endcsname{1}%
\else
    \expandafter\endinput
\fi
```

`\gobbletex@catcode`

```
\expandafter\edef\csname gobbletex@catcode\endcsname{%
    \catcode`\noexpand\@=\the\catcode`\@%
}%
\catcode`\@=11
```

`\@gobble`

`\@gobbletwo`

`\@gobblethree`

`\@gobblefour`

```
\long\def\@gobble#1{%
\long\def\@gobbletwo#1#2{%
\long\def\@gobblethree#1#2#3{%
\long\def\@gobblefour#1#2#3#4{%
```

`\@gobbleopt`

```
\long\def\@gobbleopt{%
    \@ifnextchar[%
        {\@gobbleopt@}%
        {}%
}%
```

`\@gobbleopt@`

```
\def\@gobbleopt@[#1]{}%
```

`\@gobbletwoopt`

```
\long\def\@gobbletwoopt{%
    \@ifnextchar[%
        {\@gobbletwoopt@}%
        {}%
}%
```

`\@gobbletwoopt@`

```
\def\@gobbletwoopt@[#1]{\@gobbleopt}%
```

`\@gobbleallopt`

```
\long\def\@gobbleallopt{%
    \@ifnextchar[%
        {\@gobbleallopt@}%
        {}%
}%
```

`\@gobbleallopt@`

```
\long\def\@gobbleallopt@[#1]{%
    \@ifnextchar[%
        {\@gobbleallopt@}%
        {}%
}%
```

`\@gobbletwoopttwo`

```
\long\def\@gobbletwoopttwo{%
    \@ifnextchar[%
        {\@gobbletwoopttwo@}%
        {\@gobbletwo}%
}%
```

`\@gobbletwoopttwo@`

```
\long\def\@gobbletwoopttwo@[#1]{%
    \@ifnextchar[%
        {\@gobbletwoopttwo@@}%
        {\@gobbletwo}%
}%
```

`\@gobbletwoopttwo@@`

```
\long\def\@gobbletwoopttwo@@[#1]{%
    \@gobbletwo
}%
```

`\@firstofone`

```
\long\def\@firstofone#1{#1}%
```

`\@firstoftwo`

`\@secondoftwo`

```
\long\def\@firstoftwo#1#2{#1}%
\long\def\@secondoftwo#1#2{#2}%
```

`\@firstofthree`

`\@secondofthree`

`\@thirdofthree`

```
\long\def\@firstofthree#1#2#3{#1}%
\long\def\@secondofthree#1#2#3{#2}%
\long\def\@thirdofthree#1#2#3{#3}%
```

`\@gobbletofi`

`\@gobbletoelse`

`\@gobbletoor`

```
\long\def\@gobbletofi#1#2\fi{\fi#1}%
\long\def\@gobbletoelse#1#2\else{\else#1}%
\long\def\@gobbletoor#1#2\or{\or#1}%

\gobbletex@catcode

\expandafter\ifx\csname gobble-user.tex loaded\endcsname\relax
    \expandafter\def\csname gobble-user.tex loaded\endcsname{1}%
\else
    \expandafter\endinput
\fi

\input{gobble}
```

`\gobbletex@catcode`

```
\expandafter\edef\csname gobbletex@catcode\endcsname{%
    \catcode`\noexpand\@=\the\catcode`\@%
}%
\catcode`\@=11
```

`\gobble`

`\gobbletwo`

`\gobblethree`

`\gobblefour`

```
\let\gobble\@gobble
\let\gobbletwo\@gobbletwo
\let\gobblethree\@gobblethree
\let\gobblefour\@gobblefour
```

`\gobbleopt`

```
\let\gobbleopt\@gobbleopt
```

`\gobbletwoopt`

```
\let\gobbletwoopt\@gobbletwoopt
```

`\gobbleallopt`

```
\let\gobbleallopt\@gobbleallopt
```

`\gobbletwoopttwo`

```
\let\gobbletwoopttwo\@gobbletwoopttwo
```

\firstofone

```
\let\firstofone\@firstofone
```

\firstoftwo

\secondoftwo

```
\let\firstoftwo\@firstoftwo
\let\secondoftwo\@secondoftwo
```

\firstofthree

\secondofthree

\thirdofthree

```
\let\firstofthree\@firstofthree
\let\secondofthree\@secondofthree
\let\thirdofthree\@thirdofthree
```

\gobbletofi

\gobbletoelse

\gobbletoor

```
\let\gobbletofi\@gobbletofi
\let\gobbletoelse\@gobbletoelse
\let\gobbletoor\@gobbletoor

\gobbletex@catcode
```