The `xpunctuate` package for LaTeX2e

(v2.0, last revision 2023/08/13)

Philip G. Ratcliffe

philip.ratcliffe@uninsubria.it

Dipartimento di Scienza e Alta Tecnologia
Università degli Studi dell'Insubria—Como
and
Istituto Nazionale di Fisica Nucleare
Sezione di Milano Bicocca

Abstract

The `xpunctuate` package provides package writers and ordinary users with automatic post-macro punctuation insertion, *i.e.* beyond (but similar to) that of the `xspace` package, which is required here. Three new macros are defined: `\xperiod`, `\xcomma` and `\xperiodcomma`, designed to insert the required punctuation marks *if and only if* necessary, following a similar procedure to the `xspace` package. A further four macros (`\xspaceafter`, `\xperiodafter`, `\xcommaafter` and `\xperiodcommaafter`) are constructed to avoid the small superfluous space after possible font changes (such as `\emph`) that involve the correction `\/`, which would otherwise be hidden.

1 Introduction

The present package is mainly intended for package writers, but may also serve the ordinary user; it provides additional post-macro punctuation insertion, similar to that of the `xspace` package. Three new macros are defined: `\xperiod`, `\xcomma` and `\xperiodcomma`, which, in an analogous fashion to the standard `\xspace` macro, insert any relevant punctuation where necessary. Four additional macros, of the form `\x...after`, avoid the small superfluous space due to the following period or comma being isolated from a preceding `\/`.

2 Package call

`package call` The package is loaded via a standard call:

```
\usepackage{xpunctuate}
```

2.1 User options

`user options` There are at present *no* user options.

2.2 User macros

Three user macros are defined, each having a pre-punctuation space-correcting variant, plus a similar extension of the `\xspace` macro.

`\xperiod` The purpose of this macro is to insert a period if not found as the successive LaTeX input token. Typical use will be in defining abbreviations, where there may or may not be a following *accidental* sentence-terminating full stop. If *no* explicit period follows, one is inserted, the occurrence is assumed to be mid-sentence and therefore normal inter-word spacing is applied. However, if it *is* followed by a period, then this is considered to be a sentence terminator, it is left as is with no further insertions and the appropriate trailing space will then naturally ensue.

After defining `\newcommand{\gb}{G.B\xperiod}`, the command `\gb` will determine when to insert a period after itself and when not, with also the correct mid-sentence or end-of-sentence spacing as required. Thus, the input

```
\gb is a very nice place to live.\\
I live in \gb. It is a small island off the coast of France
\gb\footnote{The small island off the coast of France.}
is a very nice place to live.
```

results in the output

G.B. is a very nice place to live.
I live in G.B. It is a small island off the coast of France.
G.B.[a] is a very nice place to live.

`\xcomma` This macro similarly inserts a comma if not found as the next token. A typical application might be following an abbreviation such as 'eg' (*i.e.* when used *without* periods), which according to different standard style manuals should or should not be followed by a comma. It has no special spacing behaviour.

`\xperiodcomma` This macro inserts a period *and* a comma if not found as the next input tokens. Typical use will be, as above, for abbreviations such as 'e.g.' (*i.e.* *with* periods) when they should be followed by a comma, but may also occur fortuitously immediately preceding an explicit sentence-closing period or indeed an explicit comma, the correct trailing space of which would then ensue. Note that this action *cannot* be reproduced via successive use of the single macros, *i.e.* the combination `\xperiod\xcomma`.

`\xspaceafter` `\xperiodafter` `\xcommaafter` `\xperiodcommaafter` The variants `\x...after` are similar to the above except that they take the object word or phrase to be spaced or punctuated as an argument; the slight incorrect spacing adjustment between the word and following period or comma is thus avoided when, for example, the word is `\emph`'asised. Other punctuation marks do *not* suffer from this problem.

Note that although the action of `\xperiodcommaafter` may also be obtained via suitable nesting of `\xperiodafter` and `\xcommaafter`, this combination has not been thoroughly tested; the compound macro is thus included for both safety and backwards compatibility.

In the case of the macros that guarantee a trailing period (*with* or *without* a comma), if the final letter of the word or phrase is a capital (*e.g.* 'Ph.D.'), to guarantee the correct end-of-sentence trailing spacing, it should be immediately followed by an explicit `\@`. That is, the definition should be of the form

[a] The small island off the coast of France.

```
\newcommand\PhD{Ph.D\@\xperiod}
```
or
```
\newcommand\PhD{\xperiodafter{Ph.D\@}}
```

N.B. If the ending capital letter or letters are *e.g.* `\emph`'asised, the `\@` should be placed *inside* the `\emph`'asis:[b]

```
\newcommand\PhD{\emph{Ph}.\emph{D\@}\xperiod}
```
or
```
\newcommand\PhD{\xperiodafter{\emph{Ph}.\emph{D\@}}}
```

The following are various other examples of possible usage:

```
\newcommand\etc{etc\xperiod}
\newcommand\eg{i.e\xperiodcomma}
\newcommand\etal{\xperiodafter{\emph{et al}}}
\newcommand\ie{\xperiodcommaafter{\emph{e}.\emph{g}}}
```

Note how the closing period in the above is *not* explicitly present, being inserted by the relevant macro as and when necessary, with too the correct spacing.

2.3 Caveats

`caveats` No specific care should be necessary when employing the macros defined here; in particular, they are all *robust*. However, trailing punctuation hidden inside or by other macro definitions may not always be correctly interpreted. Moreover, the known limitations of `\xspace` remain.

2.4 External package requirements

`xspace` The `xspace` package is required and is loaded automatically. Note that, in the *absence* of `\xspace`, the necessary presence of explicit trailing backslashes for correct spacing following command strings, would frustrate the functionality of all the macros defined here.[c]

2.5 Package conflicts

`conflicts` There are no known conflicts with any standard LaTeX2e packages.

3 Implementation

3.1 External packages

`packages` First, load the `xspace` package, for automatic trailing-space insertion:

```
1 \RequirePackage{xspace}
```

`exceptions` and add some useful extra `xspace` exceptions relevant to the present package.

```
2 \xspaceaddexceptions{\xcomma\xperiod\xperiodcomma}
```

[b] It is for this reason that it *cannot* be incorporated directly into the `xpunctuate` macros.
[c] Users who shun the use of `xspace` will probably not appreciate the present package either.

3.2 User commands

`\xspaceafter` This macro takes one argument, which it prints as is. It then inserts `\xspace` immediately after, but nothing if followed by a comma or period.

```
\NewDocumentCommand\xspaceafter{}{\xpnct@aux\xspace@aux}
```

The correct spacing between the argument and any following comma or period is thus maintained in the case of, say, an `\emph`'asised argument. The necessity arises as the presence of `\xspace` 'hides' any following punctuation from a possible `\/` spacing-correction command.

`\xperiod` The following macro inserts a period if this is not found to be the next character.

```
\NewDocumentCommand\xperiod{}{\xpnct@aux\xperiod@aux{}}
```

It may thus be used to construct common abbreviations (such as 'etc.').

`\xperiodafter` This macro takes one argument and inserts a period after it if this is not found to be the next character.

```
\NewDocumentCommand\xperiodafter{}{\xpnct@aux\xperiod@aux}
```

The correct spacing between the argument and period is again thus maintained in the case of, say, `\emph`, as in often *italicised* abbreviations (such as '*etc.*').

`\xcomma` The following macro inserts a comma if this is not found to be the next character.

```
\NewDocumentCommand\xcomma{}{\xpnct@aux\xcomma@aux{}}
```

It may thus be used to construct those common abbreviations (without periods) that might normally be followed by a comma, such as 'eg' or 'ie' according to standard American-English usage.

`\xcommaafter` The following macro takes one argument and inserts a comma after it if this is not found to be the next character.

```
\NewDocumentCommand\xcommaafter{}{\xpnct@aux\xcomma@aux}
```

`\xperiodcomma` The following macro first adds a period and then a comma if these are not found to be the next characters.

```
\NewDocumentCommand\xperiodcomma{}{\xpnct@aux\xperiodcomma@aux{}}
```

It may thus be used to construct those common abbreviations that might normally be followed by a comma (such as 'e.g.,'), the comma being inserted *if and only if* there is no following explicit sentence-ending period. Here, there is of course no problem of spacing either preceding or following the period. If only a period is found, it is treated as an end-of-sentence and the correct spacing will ensue accordingly. A following period–comma combination, if found, will be left as is.

`\xperiodcommaafter` This macro takes one argument and inserts a period and/or a comma after it if these are not found to be the next characters.

```
\NewDocumentCommand\xperiodcommaafter{}{\xpnct@aux\xperiodcomma@aux}
```

It may thus be used to construct those common *italicised* abbreviations that should normally be followed by a comma (such as '*e.g.*,').

3.3 Internal macros

`\xpnct@tok` A token register is defined to store a possible object word or phrase:

```
\newtoks\xpnct@tok
```

`\xpnct@aux` The general setup and call to the final output macros for the above user commands is handled by the following auxiliary macro:

```
\def\xpnct@aux#1#2{%
  \xpnct@tok{#2}%
  \futurelet\xpnct@nxt#1}
```

Note that all the user commands defined earlier terminate by inserting an `\xspace` *if* and *only if* the required punctuation has actually been inserted by the macro.

`\xspace@aux` Testing and output for `\xspaceafter` are performed by the following macro:

```
\def\xspace@aux{%
  \ifx\xpnct@nxt,\else
  \ifx\xpnct@nxt.\else
    \xpnct@tok\expandafter{\the\xpnct@tok\xspace}%
  \fi\fi
  \the\xpnct@tok}
```

Only periods and commas need be excluded, as the spacing is correct for all other punctuation marks.

`\xperiod@aux` Testing and output for `\xperiod` and `\xperiodafter` are performed by this macro:

```
\def\xperiod@aux{%
  \ifx\xpnct@nxt.\else
    \xpnct@tok\expandafter{\the\xpnct@tok.\@\xspace}%
  \fi
  \the\xpnct@tok}
```

The approach adopted is to treat an explicit following period as an end-of-sentence full stop; the trailing space is then handled accordingly. However, when an explicit period is *not* found), the placement of `\@` immediately after the inserted period thus avoids standard end-of-sentence spacing, assuming such an occurrence to be mid-sentence (as for abbreviations). If, in contrast, an explicit period *is* found, it is retained as is and will thus generate the correct end-of-sentence spacing.

`\xcomma@aux` Testing and output for `\xcomma` and `\xcommaafter` are performed by the following macro (shamelessly copied from an *old* `xspace.sty` and suitably hacked):

```
\def\xcomma@aux{%
  \ifx\xpnct@nxt.\else
  \ifx\xpnct@nxt,\else
  \ifx\xpnct@nxt:\else
  \ifx\xpnct@nxt;\else
  \ifx\xpnct@nxt!\else
  \ifx\xpnct@nxt?\else
  \ifx\xpnct@nxt/\else
  \ifx\xpnct@nxt-\else
  \ifx\xpnct@nxt'\else
  \ifx\xpnct@nxt)\else
  \ifx\xpnct@nxt]\else
  \ifx\xpnct@nxt\}\else
```

```
\ifx\xpnct@nxt\egroup\else
\ifx\xpnct@nxt\/\else
  \xpnct@tok\expandafter{\the\xpnct@tok,\xspace}%
\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi
\the\xpnct@tok}
```

Note that certain of the default exceptions set by the `xspace` package are missing here, for obvious reasons.

`\xperiodcomma@aux` Testing and output for `\xperiodcomma` and `\xperiodcommaafter` are performed by the following:

```
\def\xperiodcomma@aux{%
  \ifx\xpnct@nxt.\else
    \xpnct@tok\expandafter{\the\xpnct@tok.\@\xcomma}%
  \fi
  \the\xpnct@tok}
```

The presence or absence of an explicit following period is again taken to imply an end-of- or mid-sentence respectively, unless followed by an explicit comma, other closing punctuation or brackets say, when it will be assumed mid-sentence. Note, moreover, that `\@` is *still* necessary here as the period may not necessarily be immediately followed by a comma, *e.g.* if it is followed by a closing bracket or quotation marks.

`\endinput` Finally, explicitly terminate the package input here.

```
\endinput
```

Change History

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.