The xpunctuate[*] package for LaTeX2e

Philip G. Ratcliffe[†]
Dipartimento di Scienze e Alta Tecnologia
Università degli Studi dell'Insubria—Como

Abstract

This package affords the user or package writer post-macro punctuation insertion, *i.e.*, beyond (but similar to) that of the `xspace` package. Three new commands are defined: `\xperiod`, `\xcomma` and `\xperiodcomma`, which, following a similar procedure to the standard `\xspace` macro, are designed to insert the relevant punctuation *if and only if* necessary.

1 Introduction

The present package is mainly intended for package writers and provides additional post-macro punctuation insertion, similar to that of the `xspace` package. Three new commands are defined: `\xperiod`, `\xcomma` and `\xperiodcomma`, which, in an analogous fashion to the standard `\xspace` macro, insert the relevant punctuation where necessary.

2 Usage

The package is loaded via a standard package call: `\usepackage{xpunctuate}`. There are at present no user options.

2.1 User commands

The package defines three user commands, each having two variants.

`\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 period. The definition of `\xperiod` is such that if it is followed by a period, then this is considered as a sentence terminator and the appropriate trailing space is inserted. However, when no explicit period follows, the occurrence is assumed to be mid-sentence and therefore normal inter-word spacing is used.

`\xcomma` The purpose of this macro is to insert a comma if not found as the next token.

[*]This file has version number v1.0, and revision date 2012/05/21.

[†]E-mail: philip.ratcliffe@uninsubria.it

Typical use will be following an object such as "*e.g.*", which according to certain standard style manuals should be followed by a comma. This command has no special spacing behaviour.

`\xperiodcomma` The purpose of this macro is to insert a period *and* comma if not found as the next input tokens. Typical use will be, as above, following an object such as "*e.g.*", which, according to certain style manuals, should be followed by a comma, but may also occur fortuitously immediately preceding an explicit sentence-closing period, the correct trailing space of which would then be inserted.

`\xperiodafter` `\xcommaafter` `\xperiodcommaafter` These variants are similar to the above macros except that they take the word or words to be punctuated as an argument; this avoids incorrect spacing adjustment when the word is, for example, `\emph`'asised.

Note that the action of `\xperiodcommaafter` may also obtained by suitably nesting `\xperiodafter` and `\xcommaafter` though this has not been thoroughly tested; it is thus included for safety and backward compatibility.

The following are examples of possible usage:

```
\DeclareRobustCommand\etal{\xperiodafter{\emph{et al}}}
\DeclareRobustCommand\eg{e.g\xperiodcomma}
\DeclareRobustCommand\eg{\xcommaafter{\xperiodafter{\emph{e.g}}}}
```

2.2 Caveats

No particular care should be necessary in using the commands defined by this package. However, trailing punctuation hidden inside macro definitions may not be correctly interpreted.

2.3 External package requirements

The `xspace` package is required and is loaded automatically.

2.4 Package conflicts

There are no known conflicts with any standard LaTeX2e packages.

3 Implementation

3.1 External package requirements

Load the `xspace` package for automatic trailing space:

```
\RequirePackage{xspace}
```

3.2 User commands

`\xperiod` The following macro inserts a period if this is not found to be the next character. It may thus be used to construct common abbreviations (such as "*etc.*").

```
\DeclareRobustCommand\xperiod{\xprd@Set{}}
```

`\xperiodafter` This macro takes one argument and places a period after it if this is not found to be the next character. The correct spacing between the word and period is thus maintained in the case of, say, `\emph`.

```
\DeclareRobustCommand\xperiodafter[1]{\xprd@Set{#1}}
```

`\xcomma` The following macro inserts a comma if this is not found to be the next character. Thus, it may be used to construct common abbreviations and expressions that should normally be followed by a comma (such as "*e.g.*").

```
\DeclareRobustCommand\xcomma{\xcmm@Set{}}
```

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

```
\DeclareRobustCommand\xcommaafter[1]{\xcmm@Set{#1}}
```

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

```
\DeclareRobustCommand\xperiodcomma{\xpcm@Set{}}
```

It may thus be used to construct common abbreviations that should normally be followed by a comma (such as "*e.g.*"). The comma is inserted if only and if the following character does not imply the end of a period. Here, of course, there is no problem of spacing either preceding or following the period. If only a period is found this is treated as an end-of-sentence and the spacing is handled accordingly.

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

```
\DeclareRobustCommand\xperiodcommaafter[1]{\xpcm@Set{#1}}
```

3.3 Internal macros

`\xprd@Set` The setup for `\xperiod` and `\xperiodafter` is performed by the following auxiliary macro:

```
\newcommand\xprd@Set[1]{\def\xprd@Obj{#1}\futurelet\xprd@Nxt\xprd@Fin}
```

`\xprd@Fin` The testing and final output for `\xperiod` and `\xperiodafter` is made by the following auxiliary macro:

```
\newcommand\xprd@Fin{%
  \ifx\xprd@Nxt.\relax
    \let\xprd@Out\xprd@Obj
  \else
    \def\xprd@Out{\xprd@Obj.\@\xspace}%
  \fi
  \xprd@Out
}
```

Note the insertion of "`\@`" following the period when this last is not found; this avoids the standard default end-of-sentence spacing, assuming the occurrence in such a case to be mid-sentence.

`\xcmm@Set` The setup for `\xcomma` and `\xcommaafter` is performed by the following auxiliary macro:

```
\newcommand\xcmm@Set[1]{\def\xcmm@Obj{#1}\futurelet\xcmm@Nxt\xcmm@Fin}
```

`\xcmm@Fin` The testing and final output for `\xcomma` and `\xcommaafter` is made by the following auxiliary macro (shamelessly copied from an *old* `xspace` and hacked):

```
\newcommand\xcmm@Fin{%
  \let\xcmm@Out\xcmm@Obj
  \ifx\xcmm@Nxt\bgroup\else
  \ifx\xcmm@Nxt\egroup\else
  \ifx\xcmm@Nxt\/\else
  \ifx\xcmm@Nxt~\else
  \ifx\xcmm@Nxt.\else
  \ifx\xcmm@Nxt!\else
  \ifx\xcmm@Nxt,\else
  \ifx\xcmm@Nxt:\else
  \ifx\xcmm@Nxt;\else
  \ifx\xcmm@Nxt?\else
  \ifx\xcmm@Nxt/\else
  \ifx\xcmm@Nxt'\else
  \ifx\xcmm@Nxt)\else
  \ifx\xcmm@Nxt]\else
  \ifx\xcmm@Nxt-\else
    \def\xcmm@Out{\xcmm@Obj,\xspace}%
  \fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi
  \xcmm@Out
}
```

Note that there are fewer options than in the `xspace` package, for obvious reasons.

`\xpcm@Set` The setup for `\xperiodcomma` and `\xperiodcommaafter` is performed by the following auxiliary macro:

```
\newcommand\xpcm@Set[1]{\def\xpcm@Obj{#1}\futurelet\xpcm@Nxt\xpcm@Fin}
```

`\xpcm@Fin` The testing and final output for `\xperiodcomma` and `\xperiodcommaafter` is made by the following auxiliary macro:

```
\newcommand\xpcm@Fin{%
  \ifx\xpcm@Nxt.\relax
    \let\xpcm@Out\xpcm@Obj
  \else
    \def\xpcm@Out{\xpcm@Obj.\xcomma}%
  \fi
  \xpcm@Out
}
```

The choice made is that if only a following period is found, then it is treated as an end-of-sentence and the trailing space is handled accordingly.

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.