The **xpunctuate** package for \LaTeX{}2e

(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: \texttt{\xperiod}, \texttt{\xcomma} and \texttt{\periodcomma}, designed to insert the required punctuation marks *if and only if* necessary, following a similar procedure to the **xspace** package. A further four macros (\texttt{\spaceafter}, \texttt{\periodafter}, \texttt{\commaafter} and \texttt{\periodcommaafter}) are constructed to avoid the small superfluous space after possible font changes (such as \texttt{\emph}) that involve the correction \texttt{/}, 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: \texttt{\xperiod}, \texttt{\xcomma} and \texttt{\periodcomma}, which, in an analogous fashion to the standard **xspace** macro, insert any relevant punctuation where necessary. Four additional macros, of the form \texttt{\x...after}, avoid the small superfluous space due to the following period or comma being isolated from a preceding \texttt{/}.

2 Package call

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

\begin{verbatim}
\usepackage{xpunctuate}
\end{verbatim}

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 \texttt{\textbackslash xspace} macro.

\begin{description}
\item[\texttt{\textbackslash 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 \emph{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 \emph{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 \texttt{\newcommand{\textbackslash gb}{G.B\textbackslash xperiod}}, the command \textbackslash 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

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

results in the output

\begin{quote}
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.\footnote{The small island off the coast of France.} is a very nice place to live.
\end{quote}

\item[\texttt{\textbackslash 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’ (\textit{i.e.} when used \textit{without} periods), which according to different standard style manuals should or should not be followed by a comma. It has no special spacing behaviour.

\item[\texttt{\textbackslash xperiodcomma}] This macro inserts a period \textit{and} a comma if not found as the next input tokens. Typical use will be, as above, for abbreviations such as ‘e.g.’ (\textit{i.e.} \textit{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 \textit{cannot} be reproduced via successive use of the single macros, \textit{i.e.} the combination \texttt{\xperiod\xcomma}. The variants \texttt{\xspaceafter} 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 \texttt{\emph{\textbackslash emph}ased}. Other punctuation marks do \textit{not} suffer from this problem.

Note that although the action of \texttt{\xperiodcommaafter} may also be obtained via suitable nesting of \texttt{\periodafter} and \texttt{\commaafter}, 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 (\textit{with} or \textit{without} a comma), if the final letter of the word or phrase is a capital (\textit{e.g.} ‘Ph.D.’), to guarantee the correct end-of-sentence trailing spacing, it should be immediately followed by an explicit \texttt{\@}. That is, the definition should be of the form

\footnote{The small island off the coast of France.}
or \newcommand\PhD{\text{Ph.D}\mathpunct{\@}}}

N.B. If the ending capital letter or letters are \emph{asised}, the \@ should be placed \emph{inside} the \emph{asis}:

\newcommand\PhD{\emph{Ph.\emph{D}\mathpunct{\@}}} or \newcommand\PhD{\mathpunct{\@}\text{after}(\emph{Ph.\emph{D}\mathpunct{\@}})}

The following are various other examples of possible usage:

\newcommand\etc{etc\mathpunct{\@}} \newcommand\eg{i.e\mathpunct{\@}} \newcommand\etal{\text{et al}\mathpunct{\@}} \newcommand\ie{\text{e.g}\mathpunct{\@}}

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

### 2.3 Caveats

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

### 2.4 External package requirements

\texttt{xspace} The \texttt{xspace} package is required and is loaded automatically. Note that, in the \textit{absence} of \texttt{xspace}, the necessary presence of explicit trailing backslashes for correct spacing following command strings, would frustrate the functionality of all the macros defined here.

### 2.5 Package conflicts

\textit{conflicts} There are no known conflicts with any standard \LaTeX\ packages.

### 3 Implementation

#### 3.1 External packages

\textit{packages} First, load the \texttt{xspace} package, for automatic trailing-space insertion:

1 \texttt{\RequirePackage{xspace}}

\textit{exceptions} and add some useful extra \texttt{xspace} exceptions relevant to the present package.

2 \texttt{xspaceaddexceptions{,\texttt{xcomma},\texttt{period},\texttt{xperiodcomma}}}
3.2 User commands

\texttt{\textbackslash spaceafter} This macro takes one argument, which it prints as is. It then inserts \texttt{\textbackslash space} immediately after, but nothing if followed by a comma or period.

\begin{verbatim}
3 \NewDocumentCommand{\spaceafter}{}\{\xpnct@aux\spaceafter@aux\}
\end{verbatim}

The correct spacing between the argument and any following comma or period is thus maintained in the case of, say, an \texttt{\textbackslash emph}’asised argument. The necessity arises as the presence of \texttt{\textbackslash space} ‘hides’ any following punctuation from a possible \texttt{\textbackslash /} spacing-correction command.

\texttt{\textbackslash period} The following macro inserts a period if this is not found to be the next character.

\begin{verbatim}
4 \NewDocumentCommand{\period}{}\{\xpnct@aux\period@aux\}
\end{verbatim}

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

\texttt{\textbackslash periodafter} This macro takes one argument and inserts a period after it if this is not found to be the next character.

\begin{verbatim}
5 \NewDocumentCommand{\periodafter}{}\{\xpnct@aux\period@aux\}
\end{verbatim}

The correct spacing between the argument and period is again thus maintained in the case of, say, \texttt{\textbackslash emph}, as in often \textit{italicised} abbreviations (such as ‘etc.’).

\texttt{\textbackslash comma} The following macro inserts a comma if this is not found to be the next character.

\begin{verbatim}
6 \NewDocumentCommand{\comma}{}\{\xpnct@aux\comma@aux\}
\end{verbatim}

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.

\texttt{\textbackslash commaafter} The following macro takes one argument and inserts a comma after it if this is not found to be the next character.

\begin{verbatim}
7 \NewDocumentCommand{\commaafter}{}\{\xpnct@aux\comma@aux\}
\end{verbatim}

\texttt{\textbackslash periodcomma} The following macro first adds a period and then a comma if these are not found to be the next characters.

\begin{verbatim}
8 \NewDocumentCommand{\periodcomma}{}\{\xpnct@aux\periodcomma@aux\}
\end{verbatim}

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 \textit{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.

\texttt{\textbackslash periodcommaafter} 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.

\begin{verbatim}
9 \NewDocumentCommand{\periodcommaafter}{}\{\xpnct@aux\periodcommaafter@aux\}
\end{verbatim}

It may thus be used to construct those common \textit{italicised} abbreviations that should normally be followed by a comma (such as ‘\textit{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 
\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\}
Note that certain of the default exceptions set by the xspace package are missing here, for obvious reasons.

\periodcommaaux Testing and output for \periodcomma and \periodcommaafter are performed by the following:
\def\periodcommaaux{%\ifx\pnct@nxt.\else\fi\the\pnct@tok.\@latexcomma\fi\the\pnct@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.
48 \endinput

\section*{Change History}

v1.0 General: first public release \ldots 1 revamped all macros \ldots ... 4
packages: useful xspace exceptions added \ldots ... 3

v2.0 General: general upgrade \ldots 1 \@latexafter: macro added \ldots ... 4

\section*{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.

Symbols
\begin{center}
\begin{tabular}{ll}
\verb|\periodcomma| & \ldots 39 \\
\verb|\periodcommaafter| & \ldots 48 \\
\verb|\group| & \ldots 38 \\
\verb|\endinput| & \ldots 48 \\
\verb|\emergencystretch| & \ldots 37 \\
\end{tabular}
\end{center}

E exceptions \ldots 2
package call \ldots 1
packages \ldots 1
user options \ldots 1
xspace \ldots 1

G 
\begin{center}
\begin{tabular}{ll}
\verb|caveats| & \ldots 1 \\
\verb|conflicts| & \ldots 1 \\
\end{tabular}
\end{center}

caveats (generic) \ldots 1 
conflicts (generic) \ldots 1 
caveats \ldots 1 
conflicts \ldots 1 
\NewDocumentCommand \ldots 3, 4, 5, 6, 7, 8, 9
\newtoks .............. 10 \xcommaaux .... 6, 7, 25
\xcommaafter .... 2, 7 16, 21, 26, 27
\xperiod ........ 2, 2, 4 28, 29, 30, 31
\xperiod@aux .... 4, 5, 20 32, 33, 34, 35
\xperiodafter .... 2, 5 36, 37, 38, 39, 44
\xperiodcomma .... 2, 2, 8 \xpnct@tok .... 10
\xperiodcomma@aux . 24, 40, 42, 45, 47
\xperiodcommaafter 2, 9 \xspace ........ 17, 22, 40
\xperiodcommaafter 2, 9 \xspace (generic) .... 1
\xperiodcommaafter 2, 9 \xspace@aux .... 3, 14
\xspaceaddexceptions 2
\xspaceafter .... 2, 3
\xspaceafter .... 2, 3