The abstract package*

Peter Wilson
Catholic University of America
Now at peter.r.wilson@boeing.com
2000/06/11

Abstract
The abstract package provides control over the typesetting of the abstract environment, and especially provides for a one-column abstract in a two-column paper.

Contents
1 Introduction 1
2 The abstract package 2
  2.1 Options ................................. 2
  2.2 Commands and environments ........... 2
3 The package code 4
  3.1 Preliminaries ........................... 4
  3.2 Abstracts ............................... 4

1 Introduction
Questions about how to have a one-column abstract in a two-column paper seem to pop up fairly regularly on the comp.text.tex newsgroup. While an answer based on responses on ctt is provided in the FAQ, the abstract package provides a more author-friendly means of accomplishing this. Further, additional controls are provided for the typesetting of the abstract environment in general.

This manual is typeset according to the conventions of the LATEX docstrip utility which enables the automatic extraction of the LATEX macro source files [GMS94].

Section 2 describes the usage of the abstract package and commented source code is in Section 3.

*This file has version number v1.0, last revised 2000/06/11.
2 The abstract package

The typeset format of the abstract in a report or article class document depends on the class options. The formats are:

- **titlepage** class option: The abstract heading (i.e., value of \texttt{\abstractname}) is typeset centered in a bold font; the text is set in the normal font and to the normal width.

- **twocolumn** class option: The abstract heading is typeset like an unnumbered section; the text is set in the normal font and to the normal width (of a single column).

- Default (neither of the above class options): The abstract heading is typeset centered in a small bold font; the text is set in a small font and indented like the quotation environment.

Note that the titlepage option takes precedence over the twocolumn option.

The abstract package provides handles to modify the typesetting of the abstract.

### 2.1 Options

The abstract package takes the following options.

- **original.** In a twocolumn document the package default is to use the default abstract style (i.e., centered heading and small text). This option sets the typeset format of the abstract to match that of a standard two column document (i.e., section heading and normal sized text).

- **addtotoc.** The abstract title (the value of \texttt{\abstractname}) is added to the Table of Contents.

- **number.** The abstract title will be typeset as a numbered \texttt{\chapter} or as a \texttt{\section}, depending on whether the document class supports chapters or not.

  When the number option is used with a class that provides chapters, the usual \texttt{\chapter} command is used for typesetting the title. If you do not want the word ‘Chapter’ (or its equivalent) to appear, you can use the \texttt{anonchap} package to suppress this.

### 2.2 Commands and environments

The usual advice about creating a one-column abstract in a two-column document is to write code like this:

```latex
\documentclass[twocolumn...]{...}
...
\twocolumn[
  \begin{@twocolumnfalse}
  \maketitle % need full-width title
```

\footnote{The abstract environment is not available for the book class.}
With the `abstract` package, instead you do:

```
\documentclass[twocolumn...]{...}
\usepackage{abstract}
... \\
\twocolumn[
  \maketitle % need full-width title
  \begin{onecolabstract}
    abstract text...
  \end{onecolabstract}
]
\saythanks % typesets any \thanks commands
... 
```

If you want, you can use the `onecolabstract` environment in place of the `abstract` environment — it doesn’t have to be within the optional argument of the `\twocolumn` command. In fact, `onecolabstract` internally uses `abstract` for the typesetting.

These two commands can be redefined to change the fonts used for typesetting the heading (defined via `\abstractnamefont`) of the `abstract` environment and the font for typesetting the text of the abstract, respectively. The default definitions for these are designed to mimic the traditional `abstract` typesetting. Different values are used depending on whether the document uses the `titlepage` and/or `twocolumn` options. For example, in a non-titlepage one-column paper, their definitions are:

```
\newcommand{\abstractnamefont}{\normalfont\small\bfseries}
\newcommand{\abstracttextfont}{\normalfont\small}
```

```
\abstractnamepos specifies the name of the environment in which the abstract name is typeset. It can be redefined to be one of `flushleft`, `center`, or `flushright` to give a left, centered or right aligned heading; or to any other appropriate environment which is supported by the document class or used packages.

The length `\abstitleskip` is additional vertical space (either positive or negative) that is inserted between the abstract name and the text of the abstract.

The amended version of `abstract` uses a list environment for typesetting the text. These four lengths can be changed (via `\setlength`) to adjust the left and right margins, the paragraph indentation, and the vertical skip between paragraphs. The default values depend on the document options in effect.

The (internal) command `\appendiargdef{\macro}{\{\stuff\}}` will append `\stuff` at the end of the current definition of `\macro`, where `\macro` is the name
of a macro (including the backslash) which takes a single argument. For example
the following are two equivalent definitions of \mymacro:

\newcommand{\mymacro}{#1 is an artist}
\appendixargdef{\mymacro}{ in spite of being colour blind}
% or
\newcommand{\mymacro}{#1 is an artist in spite of being colour blind}

3 The package code

To try and avoid name clashes, all the internal commands include the string @bs.

3.1 Preliminaries

Announce the name and version of the package, which requires $\LaTeX\,2\alpha$.

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{abstract}[2000/06/11 v1.0 configurable abstracts]

The following \if... commands are for implementing various options.

\if@bsonecol
\ifadd@bstotoc
\ifnumber@bs
\newif\if@bsonecol
\@bsonecoltrue
\newif\ifadd@bstotoc
\add@bstotocfalse
\newif\ifnumber@bs
\number@bsfalse

Now declare and process the options.

\DeclareOption{original}{\@bsonecolfalse}
\DeclareOption{addtotoc}{\add@bstotoctrue}
\DeclareOption{number}{\number@bstrue}
\ProcessOptions\relax

3.2 Abstracts

The original definition of the abstract environment given in \texttt{ltclasses.dtx} is:

\if@titlepage
\renewenvironment{abstract}{% 
\@titlepage
\null\vfil
\@beginparpenalty\@lowpenalty
\begin{center}%
\bfseries \abstractname
\@endparpenalty\@M
\end{center}}%
{\par\vfil\null\endtitlepage}
The next set of macros comprise the `abstract` package reimplementation of the `abstract` environment.

These two macros are for specifying the fonts for typesetting the abstract’s title and text. They are initialised for the default case (i.e., no class options).

\begin{verbatim}
17 \newcommand{\abstractnamefont}{\normalfont\small\bfseries}
18 \newcommand{\abstracttextfont}{\normalfont\small}
\end{verbatim}

\begin{verbatim}
\absnamepos \abstitleskip \abslleftindent \absrightindent \absparindent \absparsep
\end{verbatim}

The abstract’s text is typeset as a single item list, called `\bstr@ctlist`. These lengths set the left and right margin indents, the paragraph indentation, and the inter-paragraph vertical space. Their initial values are all class option-dependent.

Now set all the class option-dependent values.

\begin{verbatim}
27 \if@titlepage
\setlength{\absleftindent}{\z@}
28 \renewcommand{\abstractnamefont}{\normalfont\bfseries}
29 \renewcommand{\abstracttextfont}{\normalfont}
30 \setlength{\abstitleskip}{0em}
31 \else
32 \if@twocolumn
\end{verbatim}
Values for the `twocolumn` class option.

```latex
\if@bsonecol
    \setlength{\absein\textindent}{\leftmargin}
\else
    \setlength{\absein\textindent}{\z@}
    \renewcommand{\abstractnamefont}{\normalfont\Large\bfseries}
    \renewcommand{\abstracttextfont}{\normalfont}
    \renewcommand{\absnamepos}{\flushleft}
    \setlength{\abstitleskip}{\z@}
\fi
\else
\fi
```

Values for the default, if not already initialised.

```latex
\setlength{\absein\textindent}{\leftmargin}
\fi
\fi
```

Finally, values that apply for all cases.

```latex
\setlength{\absrightindent}{\absein\leftindent}
\AtBeginDocument{\setlength{\absparindent}{\parindent}
    \setlength{\absparsep}{\parskip}}
```

@bstr@ctlist The abstract’s text is typeset within the `@bstr@ctlist` list environment.

```latex
\newenvironment{@bstr@ctlist}{% 
    \list{}{% 
        \topsep\z@ 
        \partopsep\z@ 
        \listparindent\absparindent 
        \itemindent\listparindent 
        \leftmargin\absein\leftindent 
        \rightmargin\absrightindent 
        \parsep\absparsep}% 
    \item\relax}{% \endlist
```

@put@bsintoc This macro adds the abstract’s title to the ToC. It does nothing if the abstract is being numbered.

```latex
\command{\put@bsintoc}{% 
    \ifadd@bstotoc
        \ifnumber@bs\else
            \@ifundefined{chapter}{\addcontentsline{toc}{section}{\abstractname}}{% 
                \addcontentsline{toc}{chapter}{\abstractname}}
        \fi
    \fi
```

@num@bs This macro generates a numbered abstract heading.

```latex
\command{\num@bs}{% 
    \@ifundefined{chapter}{\section{\abstractname}}{% 
        \chapter{\abstractname}}
```

6
abstract  At last we are in position to redefine the \texttt{abstract} environment. This follows very much along the lines of the original definition, but with macros inserted at strategic points.

\begin{verbatim}
\if@titlepage
For the \texttt{titlepage} option.
\renewenvironment{abstract}{% 
\null\vfil \@beginparpenalty\@lowpenalty
Number the title, if called for, otherwise typeset the title in the specified (via \texttt{absnamepos}) environment and adjust the following vertical spacing.
\ifnumber@bs \num@bs \else
  \begin{absnamepos}
  \abstractnamefont \abstractname
  \@endparpenalty\@M
  \end{absnamepos}
  \vspace{\abstitleskip}
  \fi
Try adding the title to the ToC, then start the environment for typesetting the text.
  \put@bsintoc%
  \begin{@bstr@ctlist}\abstracttextfont%
Finish the \texttt{abstract} environment.
  \par\end{@bstr@ctlist}\vfil\null\endtitlepage
\else
The \texttt{twocolumn} option and the default case. These use the same code as any style differences are embedded in the new macros.
\renewenvironment{abstract}{% 
  \begin{absnamepos} \abstractname \end{absnamepos}
  \vspace{\abstitleskip}
  \fi
\put@bsintoc%
  \begin{@bstr@ctlist}\abstracttextfont%
  \par\end{@bstr@ctlist}
\fi
\end{verbatim}

\begin{verbatim}
\appendiargdef\appendiargdef{\macro}{Additional stuff} will add \texttt{Additional stuff} at the end of the definition of \texttt{\macro}, where \texttt{\macro} is a macro that has one argument.\footnote{Code suggestions for this were made by Michael Downes (epsmjd@ams.org) and Heiko Oberdiek (oberdiek@ruf.uni-freiburg.de) on \texttt{ctt}.}
\end{verbatim}

\begin{verbatim}
\end{verbatim}
\providecommand{\appendiargdef}[2]{\begingroup
\toks@\expandafter{#1{##1}#2}\
\edef\@bsx{\endgroup \def\noexpand#1####1{\the\toks@}}\
@bsx}
\thanks

We have to keep the contents of the \thanks commands as normally these are emptied by the \maketitle command. I do this by extending the definition of the \thanks (from \ltsect.dtx) command, so that \@bs@thanks has a copy of the contents of \@thanks.

\appendiargdef{\thanks}{%
\protected@xdef\@bs@thanks{\@bs@thanks\protect\footnotetext[\the\c@footnote]{#1}}%
\let\@bs@thanks\@empty
\saythanks

This macro typesets any \thanks commands after using onecolababstract.

\newcommand{\saythanks}{\begingroup
\renewcommand{\thefootnote}{\fnsymbol{footnote}}\@bs@thanks
\endgroup\let\@bs@thanks\@empty}

\let\@bs@thanks\@empty
\saythanks

The end of this package.

References