

The **abstract** package*

Author: Peter Wilson, Herries Press
herries dot press at earthlink dot net

Maintainer: Will Robertson
will dot robertson at latex-project dot org

2009/06/08

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	3
3	The package code	4
3.1	Preliminaries	5
3.2	Abstracts	5

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 L^AT_EX DOC-STRIP utility which enables the automatic extraction of the L^AT_EX macro source files [GMS94].

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

*This file (**abstract.dtx**) has version number v1.2a, last revised 2009/06/08.

2 The abstract package

The typeset format of the **abstract** in a **report** or **article** class¹ document depends on the class options. The formats are:

- **titlepage** class option: The abstract heading (i.e., value of `\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 `\abstractname`) is added to the Table of Contents.
- **number**. The abstract title will be typeset as a numbered `\chapter` or as a `\section`, depending on whether the document class supports chapters or not.
- **style**. You specify the style for the title.² The **addtotoc** option does not apply if you use the **style** option.
- **runin**. The abstract title is typeset as a run-in³ heading. Note that the **number** and **style** options are ignored if the **runin** option is used, and the **number** and **addtotoc** options are ignored if the **style** option is given.

When the **number** option is used with a class that provides chapters, the usual `\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 **anonchap** package to suppress this.

¹The **abstract** environment is not available for the **book** class.

²This option was suggested by Will Robertson.

³This option was suggested by R. Chandrasekhar (chandra@ee.uwa.edu.au).

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:

```
\documentclass[twocolumn...]{...}
...
\twocolumn[
  \begin{@twocolumnfalse}
    \maketitle           % need full-width title
    \begin{abstract}
      abstract text...
    \end{abstract}
  \end{@twocolumnfalse}
]
... hand make footnotes for any \thanks commands
...
```

`oncolabstract` With the `abstract` package, instead you do:

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

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

`\abstractnamefont` These two commands can be redefined to change the fonts used for typesetting
`\abstracttextfont` the heading (defined via `\abstractname`) 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}
```

<code>\abstitlestyle</code>	When the style option is used the title is set as <code>\abstitlestyle{\abstractname}</code> where the default definition is simply <code>\newcommand{\abstitlestyle}[1]{#1}</code> It is up to you to redefine <code>\abstitlestyle</code> to give any effects you want (e.g., if you want the title added to the ToC you have to include the necessary code as part of your redefinition).
<code>\absleftindent</code> <code>\absrightindent</code> <code>\absparindent</code> <code>\absparsep</code>	The amended version of <code>abstract</code> uses a list environment for typesetting the text. These four lengths can be changed (via <code>\setlength</code>) to adjust the left and right margins, the paragraph indentation, and the vertical skip between paragraphs in this environment. The default values depend on the document class options in effect.
<code>\abslabeldelim</code>	If the <code>runin</code> option is used, the heading is typeset as a run-in heading. That is, it is the first piece of text on the first line of the abstract text. The <code><text></code> as dspecified by <code>\abslabeldelim{<text>}</code> is typeset immediately after the heading. By default it is defined to do nothing, but if you wanted, for example, the <code>\abstractname</code> to be followed by a colon and some extra space you could specify <code>\abslabeldelim{:\quad}</code> .
<code>\absnamepos</code>	If the <code>runin</code> option is not used then the heading is typeset in its own environment. <code>\absnamepos</code> specifies the name of the environment in which the abstract name is typeset. It can be redefined to be one of <code>flushleft</code> , <code>center</code> , or <code>flushright</code> 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.
<code>\abstitleskip</code>	With the <code>runin</code> option a space of length <code>\abstitleskip</code> is typeset before the heading. For example, if <code>\absparindent</code> is non-zero, then: <code>\setlength{\abstitleskip}{-\absparindent}</code> will typeset the heading flush left. Without the <code>runin</code> option <code>\abstitleskip</code> is additional vertical space (either positive or negative) that is inserted between the abstract name and the text of the abstract.
<code>\appendiargdef</code>	The (internal) command <code>\appendiargdef{<macro>}{<stuff>}</code> will append <code><stuff></code> at the end of the current definition of <code><macro></code> , where <code><macro></code> is the name of a macro (including the backslash) which takes a single argument. For example the following are two equivalent definitions of <code>\mymacro</code> : <code>\newcommand{\mymacro}[1]{#1 is an artist}</code> <code>\appendiargdef{\mymacro}{ in spite of being colour blind}</code> <code>% or</code> <code>\newcommand{\mymacro}[1]{#1 is an artist in spite of being colour blind}</code>

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 L^AT_EX 2_ε.

```

1 <*usc>
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{abstract}[2009/06/08 v1.2a configurable abstracts]
4

```

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

```

\if@bsonecol
\ifadd@bstotoc 5 \newif\if@bsonecol
\ifnumber@bs 6 \@bsonecoltrue
\if@bsrunin 7 \newif\ifadd@bstotoc
\if@bsstyle 8 \add@bstotocfalse
9 \newif\ifnumber@bs
10 \number@bsfalse
11 \newif\if@bsrunin
12 \@bsruninfalse
13 \newif\if@bsstyle
14 \@bsstylefalse
15

```

Now declare and process the options.

```

16 \DeclareOption{original}{\@bsonecolfalse}
17 \DeclareOption{addtotoc}{\add@bstotoctrue}
18 \DeclareOption{number}{\number@bstrue}
19 \DeclareOption{runin}{\@bsrunintrue}
20 \DeclareOption{style}{\@bsstyletrue}
21 \ProcessOptions\relax

```

Disable the number and addtotoc option if style is specified. Make sure that the number option is disabled if runin is used (otherwise the addtotoc option may not work as expected) and also disable the style option.

```

22 \if@bsstyle\number@bsfalse\add@bstotocfalse\fi
23 \if@bsrunin\number@bsfalse\@bsstylefalse\fi
24

```

3.2 Abstracts

The original definition of the `abstract` environment given in `ltclasses.dtx` is:

```

\if@titlepage
\renewenvironment{abstract}{%
\titlepage
\null\vfil
\@beginparpenalty\@lowpenalty
\begin{center}%
\bfseries \abstractname
\@endparpenalty\@M
\end{center}}%

```

```

        {\par\vfil\null\endtitlepage}
\else
\renewenvironment{abstract}{%
\if@twocolumn
\section*{\abstractname}%
\else
\small
\begin{center}%
{\bfseries \abstractname\vspace{-.5em}\vspace{\z@}}%
\end{center}%
\quotation
\fi}
{\if@twocolumn\else\endquotation\fi}
\fi

```

The next set of macros comprise the **abstract** package reimplementa-tion of the **abstract** environment.

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

```

25 \newcommand{\abstractnamefont}{\normalfont\small\bfseries}
26 \newcommand{\abstracttextfont}{\normalfont\small}
27

```

\absnamepos **\absnamepos** specifies the environment in which the abstract’s title name will be
\abstitlekip typeset, and the length **\abstitlekip** is an adjustment to the vertical space between the title and the abstract’s text. These are initialised for the default case.

```

28 \newcommand{\absnamepos}{center}
29 \newlength{\abstitlekip} \setlength{\abstitlekip}{-0.5em}

```

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

```

\absparsep 30 \newlength{\absleftindent}
31 \newlength{\absrightindent}
32 \newlength{\absparindent}
33 \newlength{\absparsep}
34

```

\abslabeldelim The contents of **\abslabeldelim** are typeset after a run-in heading.

```

\@bslabeldelim 35 \newcommand{\abslabeldelim}[1]{\def\@bslabeldelim{#1}}
36 \abslabeldelim{}

```

\@bsrunintitle This macro typeset the run-in heading.

```

37 \newcommand{\@bsrunintitle}{%
38 \hspace*{\abstitlekip}{\abstractnamefont\abstractname\@bslabeldelim}}
39

```

Now set all the class option-dependent values.

40 \if@titlepage

Values for the titlepage class option. Note that this option takes precedence over any other option.

```
41 \setlength{\absleftindent}{\z@}
42 \renewcommand{\abstractnamefont}{\normalfont\bfseries}
43 \renewcommand{\abstracttextfont}{\normalfont}
44 \setlength{\abstitlekip}{0em}
45 \else
46 \if@twocolumn
```

Values for the twocolumn class option.

```
47 \if@bsonecol
48 \setlength{\absleftindent}{\leftmargin}
49 \else
50 \setlength{\absleftindent}{\z@}
51 \renewcommand{\abstractnamefont}{\normalfont\Large\bfseries}
52 \renewcommand{\abstracttextfont}{\normalfont}
53 \renewcommand{\absnamepos}{flushleft}
54 \setlength{\abstitlekip}{0em}
55 \fi
56 \else
```

Values for the default, if not already initialised.

```
57 \setlength{\absleftindent}{\leftmargin}
58 \fi
59 \fi
```

Finally, values that apply for all cases.

```
60 \setlength{\absrightindent}{\absleftindent}
61 \AtBeginDocument{\setlength{\absparindent}{\parindent}
62 \setlength{\absparsep}{\parskip}}
63
```

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

```
64 \newenvironment{@bstr@ctlist}{%
65 \list{}{%
66 %%\topsep \z@
67 \partopsep \z@
68 \listparindent \absparindent
69 \itemindent \listparindent
70 \leftmargin \absleftindent
71 \rightmargin \absrightindent
72 \parsep \absparsep}%
73 \item\relax}
74 {\endlist}
75
```

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

```

76 \newcommand{\put@bsintoc}{%
77   \ifadd@bstotoc
78     \ifnumber@bs\else
79       \@ifundefined{chapter}{\addcontentsline{toc}{section}{\abstractname}}{%
80         \addcontentsline{toc}{chapter}{\abstractname}}
81     \fi
82   \fi}
83
\num@bs This macro generates a numbered abstract heading.
84 \newcommand{\num@bs}{%
85   \@ifundefined{chapter}{\section{\abstractname}}{%
86     \chapter{\abstractname}}
87 }
88
\abstitlestyle When the style option is called this macro is called to typeset the abstract heading;
it should be redefined by the user to suit.
89 \newcommand*{\abstitlestyle}[1]{#1}
90
abstract At last we are in position to redefine the abstract environment. This follows
very much along the lines of the original definition, but with macros inserted at
strategic points.
91 \if@titlepage
For the titlepage option.
92 \renewenvironment{abstract}{%
93   \titlepage
94   \null\vfil
95   \@beginparpenalty\@lowpenalty
96   \if@bsrunin
97   \else
If the runin option is not used: number the title, if called for, otherwise typeset
the title in the specified (via \absnamepos) environment and adjust the following
vertical spacing.
98     \if@bsstyle
99       \abstitlestyle{\abstractname}
100    \else
101      \ifnumber@bs
102        \num@bs
103      \else
104        \begin{\absnamepos}%
105          \abstractnamefont \abstractname
106          \@endparpenalty\@M
107        \end\absnamepos%
108      %% \vspace{\abstitlekip}%
109    \fi
110  \fi

```



```

111     \vspace{\abstitleskip}%
112   \fi
    Try adding the title to the ToC, then start the environment for typesetting the text.
    If the runin option is used, typeset the heading at the start of the environment.
113   \put@bsintoc%
114   \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
    Finish the abstract environment.
115   {\par\end{@bstr@ctlist}}\vfil\null\endtitlepage}
116 \else
    The twocolumn option and the default case. These use the same code as any style
    differences are embedded in the new macros.
117 \renewenvironment{abstract}{%
118   \if@bsrunin
119   \else
120     \if@bsstyle
121       \abstitlestyle{\abstractname}
122     \else
123       \ifnumber@bs
124         \num@bs
125       \else
126         \begin{\absnamepos}\abstractnamefont\abstractname\end\absnamepos%
127   %%
128     \vspace{\abstitleskip}%
129   \fi
130   \vspace{\abstitleskip}%
131   \fi
132   \put@bsintoc%
133   \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
134   {\par\end{@bstr@ctlist}}
135 \fi
136

```

onecolabstract An environment for typesetting a single column abstract, particularly as the optional argument to the `\twocolumn` command.

```

137 \newenvironment{onecolabstract}{%
138   \begin{@twocolumnfalse}\begin{abstract}}{%
139   \end{abstract}\end{@twocolumnfalse}}
140

```

\appendiargdef `\appendiargdef{\macro}{Additional stuff}` will add `Additional stuff` at the end of the definition of `\macro`, where `\macro` is a macro that has one argument.⁴

```

141 \providecommand{\appendiargdef}[2]{\begingroup
142   \toks@{\expandafter{#1{##1}#2}}%
143   \edef\@bsx{\endgroup \def\noexpand#1####1{\the\toks@}}%

```

⁴Code suggestions for this were made by Michael Downes (epsmjd@ams.org) and Heiko Oberdiek (oberdiek@ruf.uni-freiburg.de) on ctt.

```

144 \bsx}
145
\thanks We have to keep the contents of the \thanks commands as normally these are
\@bs@thanks 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.
146 \appendiargdef\thanks}{%
147 \protected@xdef\@bs@thanks{\@bs@thanks
148 \protect\footnotetext[\the\c@footnote]{#1}}%
149 }
150 \let\@bs@thanks\@empty
151
\saythanks This macro typesets any \thanks commands after using onecolabstract.
152 \newcommand{\saythanks}{\begingroup
153 \renewcommand{\thefootnote}{\fnsymbol{footnote}}\@bs@thanks
154 \endgroup\global\let\@bs@thanks\@empty}
155
The end of this package.
156 </usc>

```

References

- [GMS94] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Addison-Wesley Publishing Company, 1994.