The file `phddoc.dtx` for use with LaTeX 2ε.[*]
It contains the code for `phddoc.cls`

Yannis Lazarides

2001/08/06
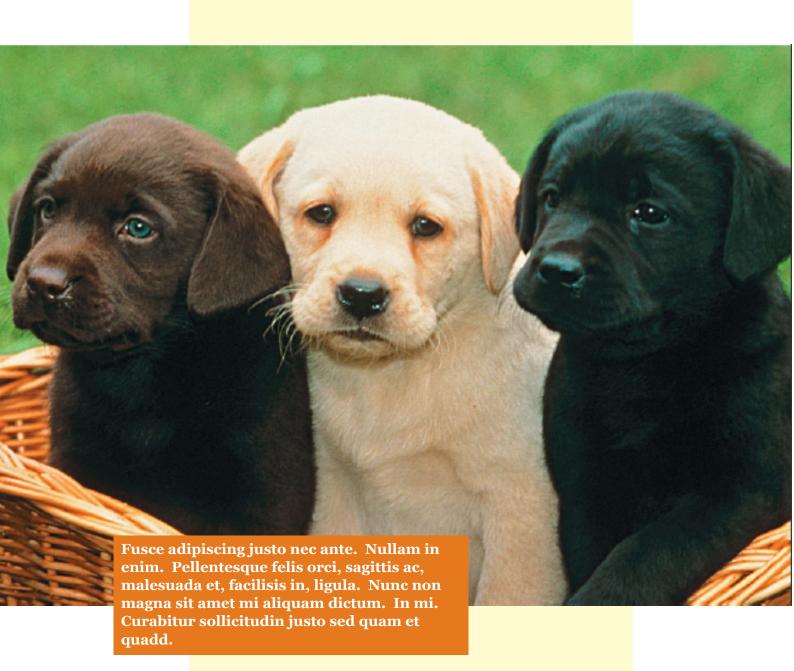
Fusce adipiscing justo nec ante. Nullam in enim. Pellentesque felis orci, sagittis ac, malesuada et, facilisis in, ligula. Nunc non magna sit amet mi aliquam dictum. In mi. Curabitur sollicitudin justo sed quam et quadd.

Contents

@tocrmarg=4em
@pnumwidth=1.5em
section
indent=1.5em
subsection
indent=3.8em
subsubsection
indent=7.0em

1 User Manual

Fusce adipiscing justo nec ante. Nullam in enim. Pellentesque felis orci, sagittis ac, malesuada et, facilisis in, ligula. Nunc non magna sit amet mi aliquam dictum. In mi. Curabitur sollicitudin justo sed quam et quadd.

Fusce adipiscing justo nec ante. Nullam in enim. Pellentesque felis orci, sagittis ac, malesuada et, facilisis in, ligula. Nunc non magna sit amet mi aliquam dictum. In mi. Curabitur sollicitudin justo sed quam et quadd.

1.1   Documentation of the LaTeX sources

This is a class for documenting the **phd** bundle, a collection of packages and classes that enables the typesetting of documents using a flexible user interface,

You may however find it generally useful as a class for typesetting the documentation of files produced in 'doc' format.

The class is written as a "self-contained" docstrip file: executing `latex phd-doc.dtx` generates the `phddoc,cls` file and typesets this documentation; execute `tex l3doc.dtx` to only generate `phddoc.cls`.

Each documented file in the standard distribution comes with extension `dtx`. The appropriate class package or initex file will be extracted from the source by the docstrip system. Each `dtx` file may be directly processed with LaTeX 2_ε, for example

```
% latex2e docclass.dtx
%
```

would produce the documentation of the Class and package interface.

Each file that is used in producing the LaTeX 2_ε format (ie not including the standard class and packages) will be printed together in one document if you LaTeX the file `sources2e.tex`. This has the advantage that one can produce a full index of macro usage across all the source files.

If you need to customise the typesetting of any of these files, there are two options:

- You can use DOCSTRIP with the module 'driver' to extract a small LaTeX file that you may edit to use whatever class or package options you require, before inputting the source file.

- You can create a file `phddoc.cfg`. This configuration file will be read whenever the `phddoc` class is used, and so can be used to customise the typesetting of all the source files, without having to edit lots of small driver files.

The second option is usually more convenient. Various possibilities are discussed in the next section.

1.2   Specification

The class builds on the **ltxdoc**[1] class and the **doc**[2] package, but since they were written many authors have come up with different ideas, as to how these documents should be produced.

The LaTeX3 Team has also more recently developed the `l3doc` class and `l3docstrip` package for documenting the l3 sources. Othe Teams such

---

[1] Carlisle, David (Mar. 2018). *The file ltxdoc.dtx for use with LaTeX2e.*
[2] Mittelbach, Frank (May 2018). *The doc and shortvrb package.*

as the developers of **pgf** prefer not to use `docstrip` and document the code and user manuals in a more traditional way, as normal documents in conjuction with external scripts written in python.

My objectives in writing this package, was to integrate the ability of the other packages in this series to document code in a flexible way. For longer books, such as a thesis, where the author might use their own developed macros, it also enables one to use such a method.

The objectives are as follows:

Flexibilty Provide flexibility to use one of the standard LaTeX 2e classes `article`, `book report` or the `KOMA` classes `scrartcl`, `scrbook`, `scrreprt` as the main class.

Style Enable the use of a fully featured key value interface for documenting the code.

Tools Provide a series of tools to create new documents, formatting and scaffolding. Currently LaTeX distributions have a plethora of tools, mostly using perl and lately l3build using Lua. Perl tools have served the community well for many years. One such tool `ctanify` does not work using normal dstributions as the Perl bundled in the distributions has some missiong modules. Go is a cross-compliation systems language enabling scripts to be bundled for different operating systems easily, hence the choice here.[3] Some of these problems with Perl on Windows can be overcome using `Strawberry Perl`[4] For any conflicts follow the guidelines in penwatch.[5]

```
phd ctanify  myclass.dtx  myclass.ins  README
```

1.3 Customisation

The simplest form of customisation is to pass more options to the `article` class which is loaded by **phddoc** . For instance if you wish all the documentation to be formated for A4 paper, add the following line to `phddoc.cfg`:

```
% \PassOptionsToClass{a4paper}{article}
%
```

All the source files are in two parts, separated by `\StopEventually`. The first part (should) contain 'user' documentation. The second part is a full

---

[3]See for example https://tex.stackexchange.com/questions/256096/which-perl-to-install-for-xindy-with-miktex-on-windows

[4]Donload at http://strawberryperl.com/. This will also enable xindy to work on a MikTeX distribution.

[5]https://www.penwatch.net/cms/pip_conflict/

documented listing of the source code. The doc package provides the command \OnlyDescription which suppresses the code listings. This may also be used in the configuration file, but as the doc package is read later, you must delay the execution of \OnlyDescription until after the doc package has been read. The simplest way is to use \AtBeginDocument. Thus you could put the following in your phddoc.cfg.

```
% \AtBeginDocument{\OnlyDescription}
%
```

If the full source listing sources2e.tex is processed, then an index and change history are produced by default, however indices are not normally produced for individual files.

As an example, consider ltclass.dtx, which contains the sources for the new class and package interface commands. With no cfg file, a 19 page document is produced. With the above configuration a slightly more readable document (4 pages) is produced.

Conversely, if you really want to read the source listings in detail, you will want to have an index. Again the index commands provided by the doc package may be used, but their execution must be delayed.

```
% \AtBeginDocument{\CodelineIndex\EnableCrossrefs}
% \AtEndDocument{\PrintIndex}
%
```
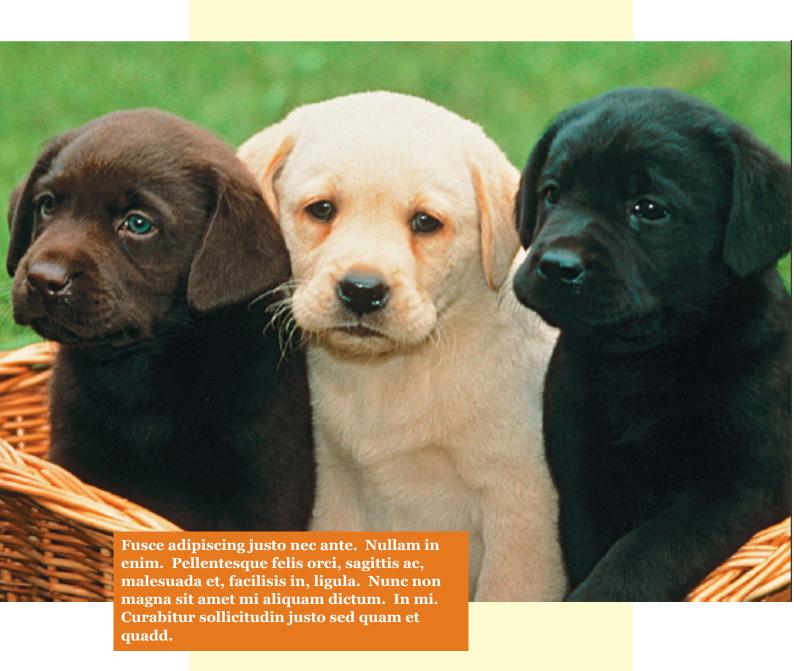
The doc package writes index files to be sorted using MakeIndex with the gind style, so one would then use a command such as

```
% makeindex -s gind.ist ltclass.idx
%
```

and re-run LaTeX.

Similarly to print a Change history, you would add

```
% \AtBeginDocument{\RecordChanges}
% \AtEndDocument{\PrintChanges}
%
```

to phddoc.cfg, and use MakeIndex with a comand such as

```
% makeindex -s gglo.ist -o ltclass.gls ltclass.glo
%
```

Finally if you do not want to list all the sections of source2e.tex, you can use \includeonly in the cfg file:

```
% \includeonly{ltvers,ltboxes}
%
```

2 Implementation Code

Fusce adipiscing justo nec ante. Nullam in enim. Pellentesque felis orci, sagittis ac, malesuada et, facilisis in, ligula. Nunc non magna sit amet mi aliquam dictum. In mi. Curabitur sollicitudin justo sed quam et quadd.

Fusce adipiscing justo nec ante. Nullam in enim. Pellentesque felis orci, sagittis ac, malesuada et, facilisis in, ligula. Nunc non magna sit amet mi aliquam dictum. In mi. Curabitur sollicitudin justo sed quam et quadd.

2.1   Options

```
 1 ⟨*class⟩
 2 \DeclareOption{a5paper}{\@latexerr{Option not supported}%
 3    {}}
 4 \newif\if@book
 5 \newif\if@article
 6 \newif\if@report
 7 \newif\if@scrbook
 8 \newif\if@scrartcl
 9 \newif\if@scrreprt
10
11 \DeclareOption{article}{\@articletrue}
12 \DeclareOption{book}{\@booktrue}
13 \DeclareOption{report}{\@reporttrue}
14 \DeclareOption{scrbook}{\@scrbooktrue}
15 \DeclareOption{scrartcl}{\@scrartcltrue}
16 \DeclareOption{scrreprt}{\@scrreprttrue}
17
18 \DeclareOption*{%
19    \PassOptionsToClass  {\CurrentOption}{book}}
```

2.2   Configuration

Input a local configuration file, if it exists.

```
20 \InputIfFileExists{phddoc.cfg}
21          {\typeout{************************************^^J%
22                    * Local config file phddoc.cfg used^^J%
23                    ************************************}}
24          {}
```

2.3   Option Processing

```
25 \ProcessOptions
```

2.4   Loading book and doc

The original phddoc uses the article class. For longer documentation it is preferable to use the book. This means we might get an error on the production of a title, which we will fix very soon.

```
26 \if@book
27   \LoadClass{book}
28     \else
29       \if@article
30         \LoadClass{article}
31         \cxset{section number prefix=,}
32         \else
33           \if@report
34             \LoadClass{report}
```

```
35          \else
36             \if@scrbook
37             \LoadClass{scrbook}
38             \else
39             \if@scrartcl
40                %\cxset{section number prefix=,}
41                \LoadClass{scrartcl}
42                \else
43                \if@scrreprt
44                \LoadClass{scrreprt}
45             \fi
46          \fi
47       \fi
48    \fi
49  \fi
50 \fi
```

```
51 \RequirePackage{doc}
52 \RequirePackage{expl3}
```

Make | be a 'short verb' character, but not in the document preamble, where an active character may interfere with packages that are loaded.

```
53 \AtBeginDocument{\MakeShortVerb{\|}}
```

As 'doc' documents tend to have a lot of monospaced material, Set up some tt substitutions to occur silently.

```
54 \DeclareFontShape{OT1}{cmtt}{bx}{n}{<-> ssub * cmtt/m/n}{}
55 \DeclareFontFamily{OMS}{cmtt}{\skewchar\font 48}  % '60
56 \DeclareFontShape{OMS}{cmtt}{m}{n}{<-> ssub * cmsy/m/n}{}
57 \DeclareFontShape{OMS}{cmtt}{bx}{n}{<-> ssub * cmsy/b/n}{}
```

This substitution is in the standard fd file, but not silent.

```
58 \DeclareFontShape{OT1}{cmss}{m}{it}{<->ssub*cmss/m/sl}{}
```

```
59 \CodelineNumbered
60 \DisableCrossrefs
```

Increase the text width slightly so that width the standard fonts 72 columns of code may appear in a macrocode environment.

```
61 \setlength{\textwidth}{355pt}
```

Increase the marginpar width slightly, for long command names. And increase the left margin by a similar amount

```
62 \addtolength\marginparwidth{30pt}
63 \addtolength\oddsidemargin{20pt}
64 \addtolength\evensidemargin{20pt}
```

```
65 \setcounter{StandardModuleDepth}{1}
```

2.5 Useful abbreviations

The **phd-documentation** provides numerous commands for typesetting LaTeX code. It is imported automatically by the `phddoc` class and hence the following macros are described here for convenience.

`\cmd{\foo}` Prints `\foo` verbatim. It may be used inside moving arguments. It can *not* be use to record commands that are defined as "`\outer`" nor is it possible to use it on conditionals such as `\iftrue` or defined by `\newif`. `\cs{foo}` also prints `\foo`, for those who prefer that syntax. (This second form can be used to record all type of commends so the above restrictions do not apply.

`\cmd`
`\cs`
```
66 \def\cmd#1{\cs{\expandafter\cmd@to@cs\string#1}}
67 \def\cmd@to@cs#1#2{\char\number`#2\relax}
68 %\newcommand\cs[1]{\color{blue}{\texttt{\char`\ #1}}}
```

`\marg` `\marg{text}` prints {⟨*text*⟩}, 'mandatory argument'.
```
69 %\providecommand\marg[1]{%
70 %  {\ttfamily\char`\{}\meta{#1}{\ttfamily\char`\}}}
```

`\oarg` `\oarg{text}` prints [⟨*text*⟩], 'optional argument'.
```
71 %\providecommand\oarg[1]{%
72 %  {\ttfamily[}\meta{#1}{\ttfamily]}}
```

`\parg` `\parg{te,xt}` prints (⟨*te,xt*⟩), 'picture mode argument'.
```
73 \providecommand\parg[1]{%
74   {\ttfamily(}\meta{#1}{\ttfamily)}}
```

2.6 DocInclude

```
75 \@addtoreset{CodelineNo}{part}
```

`\DocInclude` More or less exactly the same as `\include`, but uses `\DocInput` on a `dtx` file, not `\input` on a `tex` file.
```
76 \def\partname{File}
```
```
77 \newcommand*{\DocInclude}[1]{%
78   \relax
79   \clearpage
80   \docincludeaux
81   \IfFileExists{#1.fdd}{\def\currentfile{#1.fdd}}{\def\currentfile{#1.dtx}}%
82   \ifnum\@auxout=\@partaux
83     \@latexerr{\string\include\space cannot be nested}\@eha
84   \else \@docinclude#1 \fi}
85 \def\@docinclude#1 {\clearpage
86 \if@filesw \immediate\write\@mainaux{\string\@input{#1.aux}}\fi
87 \@tempswatrue\if@partsw \@tempswafalse\edef\@tempb{#1}\@for
88 \@tempa:=\@partlist\do{\ifx\@tempa\@tempb\@tempswatrue\fi}\fi
```

```
89 \if@tempswa \let\@auxout\@partaux \if@filesw
90 \immediate\openout\@partaux #1.aux
91 \immediate\write\@partaux{\relax}\fi
```

We need to save (and later restore) various index-related commands which
might be changed by the included file.

```
92 \let\@phddoc@PrintIndex\PrintIndex
93 \let\PrintIndex\relax
94 \let\@phddoc@PrintChanges\PrintChanges
95 \let\PrintChanges\relax
96 \let\@phddoc@theglossary\theglossary
97 \let\@phddoc@endtheglossary\endtheglossary
98 \part{\currentfile}%
99   {\let\ttfamily\relax
100   \xdef\filekey{\filekey, \thepart={\ttfamily\currentfile}}}%
101 \DocInput{\currentfile}%
102 \let\PrintIndex\@phddoc@PrintIndex
103 \let\PrintChanges\@phddoc@PrintChanges
104 \let\theglossary\@phddoc@theglossary
105 \let\endtheglossary\@phddoc@endtheglossary
106 \clearpage
107 \@writeckpt{#1}\if@filesw \immediate\closeout\@partaux \fi
108 \else\@nameuse{cp@#1}\fi\let\@auxout\@mainaux}

109 \gdef\codeline@wrindex#1{\if@filesw
110        \immediate\write\@indexfile
111            {\string\indexentry{#1}%
112            {\filesep\number\c@CodelineNo}}\fi}%

113 \let\filesep\@empty
```

\aalph   Special form of \alph as currently source2e.tex includes more than 26
         files .

```
114 \def\aalph#1{\@aalph{\csname c@#1\endcsname}}
115 \def\@aalph#1{%
116 \ifcase#1\or a\or b\or c\or d\or e\or f\or g\or h\or i\or
117        j\or k\or l\or m\or n\or o\or p\or q\or r\or s\or
118        t\or u\or v\or w\or x\or y\or z\or A\or B\or C\or
119        D\or E\or F\or G\or H\or I\or J\or K\or L\or M\or
120        N\or O\or P\or Q\or R\or S\or T\or U\or V\or W\or
121        X\or Y\or Z\else\@ctrerr\fi}
```

\docincludeaux

```
122 \def\docincludeaux{%
123   \def\thepart{\aalph{part}}\def\filesep{\thepart-}%
124   \let\filekey\@gobble
125   \g@addto@macro\index@prologue{%
126     \gdef\@oddfoot{\parbox{\textwidth}{\strut\footnotesize
127       \raggedright{\bfseries File Key:} \filekey}}%
128     \let\@evenfoot\@oddfoot}%
```

```
129    \global\let\docincludeaux\relax
130 \gdef\@oddfoot{%
131   \expandafter\ifx\csname ver@\currentfile\endcsname\relax
132    File \thepart: {\ttfamily\currentfile} %
133   \else
134   \GetFileInfo{\currentfile}%
135   File \thepart: {\ttfamily\filename} %
136   Date: \filedate\ %
137   Version \fileversion
138   \fi
139   \hfill\thepage}%
140 \let\@evenfoot\@oddfoot}%
```

ddangernote

Use the double danger bend if there is something which could cause serious problems when used in a wrong way. Better the normal user does not know about such things.

```
141 \ExplSyntaxOn
142 \newenvironment {ddangernote}
143   {
144    \begin{trivlist}\item[]\noindent
145    \begingroup\hangindent=3.5pc\hangafter=-2
146    \cs_set:Npn \par{\endgraf\endgroup}
147    \hbox to0pt{\hskip-\hangindent\dbend\kern2pt\dbend\hfill}\ignorespaces
148   }{
149      \par\end{trivlist}
150   }
151 \ExplSyntaxOff
```

This is a double bend danger sign from **manfnt**[1]

```
152 \def\task#1#2{}
153 ⟨/class⟩
```

---
[1]Kielhorn, Axel and Kosygin, Denis (July 1999). *The Manfnt package. A quick way to access the symbols in manfnt.*