

The file `phddoc.dtx` for use with \LaTeX 2 _{ϵ} .^{*}
It contains the code for `phddoc.cls`

Yannis Lazarides

2001/08/06

^{*}This file has version number v0.02, dated 2001/08/06.

Contents

```
@tocmarg=4em
@pnumwidth=2.3em
section
indent=1.5em
subsection
indent=3.8em
subsubsection
indent=7.0em
```



1 The phddoc Class	1
2 Documentation of the \LaTeX sources	1
3 Specifiction	2
4 Customisation	3
5 Options	4
6 Configuration	4
7 Option Processing	5
8 Loading book and doc	5
9 Useful abbreviations	6
10 DocInclude	6

1 The phddoc Class

2 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 phddoc.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.

3 Specification

The class builds on the `ltxdoc`¹ class and the `doc` 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. Other Teams such 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 conjunction 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:

Flexibility Provide flexibility to use `article`, `book` or `report` classes as the main class.

¹`ltxdoc`.



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.

4 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 formatted 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 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 `ltxclass.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 command 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}
%
```

5 Options

```
1 \class
2 \DeclareOption{a5paper}{\@latexerr{Option not supported}%
3   {}}
4 \newif\if@book
5 \newif\if@article
6 \newif\if@report
7 \DeclareOption{article}{\@articletrue}
8 \DeclareOption{book}{\@booktrue}
9 \DeclareOption{report}{\@reporttrue}
10 \DeclareOption*{%
11   \PassOptionsToClass {\CurrentOption}{book}}
```

6 Configuration

Input a local configuration file, if it exists.

```
12 \InputIfFileExists{phddoc.cfg}
13   {\typeout{*****^J%
14             * Local config file phddoc.cfg used^^J%
15             *****}}
16   {}
```



7 Option Processing

17 \ProcessOptions

8 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.

```
18 \if@book
19   \LoadClass{book}
20 \else
21   \if@article
22     \LoadClass{article}
23   \else
24     \LoadClass{report}
25   \fi
26 \fi
27 \RequirePackage{doc}
```

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

```
28 \AtBeginDocument{\MakeShortVerb{||}}
```

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

```
29 \DeclareFontShape{OT1}{cmtt}{bx}{n}{<-> ssub * cmmt/m/n}{}
30 \DeclareFontFamily{OMS}{cmtt}{\skewchar\font 48} % '60
31 \DeclareFontShape{OMS}{cmtt}{m}{n}{<-> ssub * cmsy/m/n}{}
32 \DeclareFontShape{OMS}{cmtt}{bx}{n}{<-> ssub * cmsy/b/n}{}

```

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

```
33 \DeclareFontShape{OT1}{cmss}{m}{it}{<->ssub*cmss/m/sl}{}
34 \CodelineNumbered
35 \DisableCrossrefs
```

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

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

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

```
37 \addtolength{\marginparwidth}{30pt}
38 \addtolength{\oddsidemargin}{20pt}
39 \addtolength{\evensidemargin}{20pt}
40 \setcounter{StandardModuleDepth}{1}
```



9 Useful abbreviations

`\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 41 \def\cmd#1{\cs{\expandafter\cmd@to@cs\string#1}}
     42 \def\cmd@to@cs#1#2{\char\number`#2\relax}
     43 \DeclareRobustCommand\cs[1]{\texttt{\char`\#1}}

\marg \marg{text} prints {<text>}, ‘mandatory argument’.
     44 \providecommand\marg[1]{%
     45   {\ttfamily\char`\}\meta{#1}{\ttfamily\char`\}}

\oarg \oarg{text} prints [<text>], ‘optional argument’.
     46 \providecommand\oarg[1]{%
     47   {\ttfamily[]\meta{#1}{\ttfamily[]}}

\parg \parg{te,xt} prints (<te,xt>), ‘picture mode argument’.
     48 \providecommand\parg[1]{%
     49   {\ttfamily{}\meta{#1}{\ttfamily{}}}
```

10 DocInclude

```
50 \@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.
51 \def\partname{File}
52 \newcommand*\DocInclude[1]{%
53   \relax
54   \clearpage
55   \docincludeaux
56   \IfFileExists{#1.fdd}{\def\currentfile{#1.fdd}}{\def\currentfile{#1.dtx}}%
57   \ifnum\@auxout=\@partaux
58     \@latexerr{\string\include\space cannot be nested}\@eha
59   \else \@docinclude#1 \fi
60 \def\@docinclude#1 {\clearpage
61 \if@filesw \immediate\write\@mainaux{\string\input{#1.aux}}\fi
62 \@tempswatrue\if@partsw \@tempswafalse\edef\@tempb{#1}\@for
```



```

63 \@tempa:=\@partlist\do{\ifx\@tempa\@tempb\@tempswattrue\fi}\fi
64 \if@tempswa \let\@auxout\@partaux \if@filesw
65 \immediate\openout\@partaux #1.aux
66 \immediate\write\@partaux{\relax}\fi

```

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

```

67 \let\@phddoc@PrintIndex\PrintIndex
68 \let\PrintIndex\relax
69 \let\@phddoc@PrintChanges\PrintChanges
70 \let\PrintChanges\relax
71 \let\@phddoc@theglossary\theglossary
72 \let\@phddoc@endtheglossary\endtheglossary
73 \part{\currentfile}%
74 {\let\ttfamily\relax
75 \xdef\filekey{\filekey, \thepart={\ttfamily\currentfile}}}%
76 \DocInput{\currentfile}%
77 \let\PrintIndex\@phddoc@PrintIndex
78 \let\PrintChanges\@phddoc@PrintChanges
79 \let\theglossary\@phddoc@theglossary
80 \let\endtheglossary\@phddoc@endtheglossary
81 \clearpage
82 \@writeckpt{#1}\if@filesw \immediate\closeout\@partaux \fi
83 \else\@nameuse{cp@#1}\fi\let\@auxout\@mainaux}

84 \gdef\codeline@wrindex#1{\if@filesw
85     \immediate\write\@indexfile
86     {\string\indexentry{#1}%
87     {\filesep\number\c@CodelineNo}}\fi}%

88 \let\filesep\@empty

```

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

```

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

```

`\docincludeaux`

```

97 \def\docincludeaux{%
98     \def\thepart{\aalph{part}}\def\filesep{\thepart-}%

```




```
99 \let\filekey\@gobble
100 \g@addto@macro\index@prologue{%
101   \gdef\@oddfoot{\parbox{\textwidth}{\strut\footnotesize
102     \raggedright{\bfseries File Key:} \filekey}}%
103   \let\@evenfoot\@oddfoot}%
104 \global\let\docincludeaux\relax
105 \gdef\@oddfoot{%
106   \expandafter\ifx\csname ver@\currentfile\endcsname\relax
107     File \thepart: {\ttfamily\currentfile} %
108   \else
109     \GetFileInfo{\currentfile}%
110     File \thepart: {\ttfamily\filename} %
111     Date: \filedate\ %
112     Version \fileversion
113   \fi
114   \hfill\thepage}%
115 \let\@evenfoot\@oddfoot}%

116 \def\task#1#2{}
117 \</class>
```