1 What is this thing you Earth people call PDCMAC?

1·1 Introduction

PDCMAC is collection of TEX definition files (macro files) which may be useful for setting documents using plain TEX ('plain' here meaning TEX formats following similar conventions to those of Appendix B of the *TEXbook*, as opposed to more complex formats like LaTEX). This package is much less powerful than LaTEX 2e; it is intended to be a simpler solution to simpler requirements. Because it is less complex, the macro code should be more readily adapted by other TEX hackers.

The package includes a font-selection system, an output routine, general formatting macros, and 'style files' which input the other files and set the format for documents.

The name 'PDCMAC' is pronounced '*p-d-c*-mac'. In file names where case matters it is always written in all-lower-case. The fairly consistent use of a '`pdc-`' or '`ma-`' prefix in this and other names is intended to prevent these files clashing with files from other packages. (The '`ma-`' files are part of the Malvern package.)

1·2 Conventions in this guide

Contents and names of computer files, and commands to be typed literally are printed in `this distinctive font`. Placeholders to be filled in with real file names etc. are written in *this italic font*.

1·3 Copying

The `tex` files generated by the `dtx` files are like 'object files'; you should not distribute them without their source files (the `dtx` files).

Do not modify the generated `tex` files; if you must modify the macros, do this by editing the `dtx` files and running them through TEX again. If you must distribute modified versions (instead of persuading me to modify my copies), help reduce the proliferation of incompatible versions by doing the following:

- describe the modifications clearly in the printed documentation;
- say who modifed them in the header comments in the definition files, and change the version identifier;
- use a different name for the modifed definition files.

The last point is so that documents using the unmodified versions can coexist with the ones using modified ones.

1·4 Feedback

I am very interested to hear from people who find a use for this package. Please send comments and suggestions, or reports of bugs, to the address above. If you find PDCMAC useful or amusing, please send me a pretty postcard. Thanks.

2 Unpacking and installing the files

2·1 Unpacking

Two common formats for archives are

- (on Unix) `tar` files, compressed using GNU `zip` (`gzip`), and
- (on MS-DOS) PKZIP-style archives.

Compressed tarfiles will have names like `pdcmac-1.0.tar.gz` or `pdcmac10.tgz` (the latter form is required by ISO-9660[1] file systems). Unpack the package with something like

```
zcat pdcmac10.tgz | tar -xf -
```

This generates a new directory called `pdcmac-1.0`.

PKZIP archives unpack files into the current directory, so they are unpacked like this:

```
md pdcmac
cd pdcmac
unzip a:\pdcmac10.zip
```

assuming the zipfile is so named.

There is a list of the files in the release in the appendix.

2·2 Generating the macro files

The macro files are packaged with their documentation in `dtx`[2] files; run plain TEX on each of the `dtx` files in turn to generate the macro files and the printed documentation. The resulting definition files have almost no comments in them; instead you must read the `dtx` files or the printed documentation.

The macro files are written in the current working directory. They are identical to the code lines in the printed documentation (they are generated from the same text in the `dtx` files).

There is a file `pdcmondo.tex` which reads all of the `dtx` files in turn except `pdcsty.dtx` and produces one large (70+ pages) document as well as all their macro files. This is most useful if you want to make a printed listing rather than keeping `dvi` files for reading online.

2·3 Installing the files

The definition (`tex`) files belong in a directory where TEX can find them. In the new soon-to-be-standardized file name conventions

[1] ISO 9660 is the standard for CD-ROM file systems. Its file names are like MS-DOS file names: a sequence of 8-letter components followed by a '`.`' and three-letter suffix. ISO 9660 names use capital letters, but on case-sensitive operating systems these are usually transliterated to lower-case.

[2] The LaTeX 2e distribution uses the file name suffix `dtx` for files with a similar function. Unlike the LaTeX system, the `dtx` files for PDCMAC produce the printed documentation and unpack the macro files themselves using a single macro file `pdccode.tex` and a single run through plain TEX; there are no `drv` or `ins` files.

TWG-TDS 0·61[3] this is the directory `$texmf/tex/plain/pdcmac/`. On older systems, the files go with all the other macro files.

With TWG-TDS 0·61 the documentation goes in `$texmf/doc/plain/pdcmac/`. If you do not have a directory for documentation, the documentation files might as well go in the TEX inputs directory as well.

2·4 Configuration on Unix systems

There is a `configure` script and makefile template included, which allows the process of unpacking to be run automatically on Unix systems.[4] The remainder of this section assumes you are installing PDCMAC on a Unix system.

2·5 Running `configure`

Start by running a Bourne Shell on the `configure` script, by typing '`sh configure`'. This examines your file system and attempts to guess suitable directories in which to put macro and documentation files. The `configure` script understands options listed in Table 1.

Table 1 Options for `configure`. Other options are ignored.

Option	*Meaning*
`-h`, `--help`	Print a summary of options
`-n`, `--no-create`	Create `config.status` but don't run it to make `makefile`.
`-t`*dir* `--texmf=`*dir*	Says where to find a TEX directory hierarchy. For example, '`-t /usr/texmf`' or '`-t/usr/local/lib/tex3.14/tex`'. The `configure` script will often guess correctly without this option.
`-p`*dir* `--prefix=`*dir*	Specifies the parent of the TEX directory, for example, '`-p/usr`' or '`-p/usr/local/lib`'. This is for compatability with the GNU coding standards.
`-wtds` `--with-tds`	Specify that the TEX directory uses some approximation to the TWG-TDS 0·61 file name conventions. This should not be necessary as `configure` will usually guess correctly.

The configuration process creates a script `config.status` which records the configuration; running `config.status` generates a file `makefile`[5] from the template `makefile.in`.

The `config.status` script has one option `-r` (or `--recheck`), which re-runs `configure` with the same arguments as were used to generate `config.status`; any options following `-r` are passed to `configure`.

[3] The TUG Working Group on a TEX Directory Structure, *A Directory Structure for Implementation-Independent TEX Files* Version 0·61 (`ftp://ftp.th-darmstadt.de/pub/tex/TDS-compliant/draft/twg-tds.dvi`, 10 February 1995).

[4] The configuration system is based on the GNU Coding Standards, but was written by hand rather than using Autoconf.

[5] Usually a makefile is called `Makefile`, but I wanted to make the package proof against file name munging from being copied onto MS-DOS discs.

2·6 Running `make`

Now you can use the `make` command to unpack and install all the files. Do '`make`' to generate all the definition files and documentation. Then '`make install`' to copy the macros into TEX's macros directory and the documentation into TEX's documentation area (assuming there is one). The standard targets which the makefile understands are listed in Table 2.

Table 2 Conventional targets defined in `makefile`.

Command	*Meaning*
`make all`	Generate all the definitions files and `dvi` files
`make install`	Generate the definition files and copy them in to TEX's macro area. Also copy the `dvi` files into TEX's documentation area, if possible.
`make uninstall`	Delete all the files that '`make install`' would install.
`make mostlyclean`	Delete some files but not as many as '`make clean`'.
`make clean`	Delete files from the current directory that are normally created by '`make all`'. Don't delete files that could be built using the makefile but which come with the distribution.
`make distclean`	Delete some more files, including those made by configuration. If you have unpacked the files and generated the macro files without creating any other files, this should leave only the files in the distribution.
`make realclean`	Delete files deleted '`make distclean`' and any others that can be rebuilt using the makefile.
`make TAGS`	Generate a tags table file for Emacs.
`make dist`	Make a tarfile and zipfile for the package.

3 Using PDCMAC style files for your documents

Normally a document will start by reading one of the style files, which in turn load the various definition files. The style files are intended to be more-or-less compatible with each other. I have arbitrarily divided the style files into 'leaflets' and 'docs'.

3·1 Leaflet styles

A leaflet is only a few pages, so does not need a table of contents or division into large units. There is still a `\section` command, but it is designed for smaller divisions than the `\section` command used in 'docs' (in a leaflet, `\section` produces a heading with prominence similar to that produced by `\subsec` in a doc).

A leaflet-style document has no front matter, and so should start with some sort of heading for the title.

```
\input pdccmlft
\majorheadline{ title } \noheadlinetrue
  commands to generate the title at the top of the first page
  text of the document, perhaps using \section commands
```

```
\bye
```

3·2 Doc styles

A doc is something larger than a leaflet but smaller than a book. It has a table of contents and numbered sections and subsections, with section titles being reproduced in the headline. There is no provision for cross-references and automatic bibliography (which require an `aux` file and at least two passes through TEX).

Sections may be grouped into larger divisions I have called ***part***s. Parts are numbered independently of sections, in upper-case roman numerals. (There is no special reason for not numbering sections within parts; I just prefer to have fewer levels of numbering, so we get '§12·6' instead of 'Subsection 4·1·6'.)

```
\input pdccmdoc
\part{ title } —or— \majorheadline{ title }
\section{ title }
  contents of section
more sections
\frontmatter
  front matter
\endfrontmatter
\bye
```

3·3 Front matter

The front matter of the document—the title page, preface, forword, etc.—must be printed *last*, with the table of contents at the end of the front matter; this is so that the table of contents may be accumulated during the TEXing of the file.[6] The front matter starts with `\frontmatter` and may contain `\section` commands. Such sections will be unnumbered and will not appear in the table of contents.

For a short document, a separate title page is probably excessive, and an abstract may be preferable to a preface. In this case the first page after `\frontmatter` could have the title of the document (with author etc.) followed by an abstract, any copyright information (or other small print), and the contents (generated by `\endfrontmatter`). In other words, something like this:

```
\frontmatter
  commands to print the title, etc.
  \abstract
    the text of the abstract
  \endabstract
  copyright information, etc.
\endfrontmatter
```

[6] It has a benefit for people reading the document with a browser: page 1 of the document is the first page of the DVI file, which makes selecting a given page easy, and the table of contents is at the very end, so the browser's 'go to last page' command can be used to find the table of contents quickly.

For a longer document, there will be a separate title page and perhaps a preface.

```
\frontmatter
  \titlepage
    commands to print the title, etc.
  \splittitlepage
    print copyright information, etc.
  \endtitlepage
  \section{Preface}
    text of preface, etc.
\endfrontmatter
```

The macro `\splittitlepage` marks the division between the title page (title recto, page i) and the back of the title page (title verso, page ii), which is where copyright information goes. When formatting for one-sided printing, the copyright information belongs on the title recto, because the title verso will be blank, so `\splittitle` instead does `\vfill`.

3·4 New symbols

Several new symbols common to Malvern A and PostScript fonts are added (listed in Table 3). Approximations built from other glyphs are available in Computer Modern documents.

Table 3 New symbols

¢	`\cents`	£	`\pounds`	¥	`\yen`	*f*	`\florin`
¤	`\currency`	«	`\lguillemet`	»	`\rguillemet`	·	`\gbdecimal`
‰	`\permille`	®	`\registered`	ª	`\orda`	º	`\ordo`
§	`\S`	¶	`\P`	†	`\dag`	‡	`\ddag`

The symbols in the last row are 'new' in the sense that they will change according to the current font when using Malvern or PostScript fonts.

The maths symbols in Table 4 will be in the current `\rm` font (fam 0) in PostScript documents.

Table 4 Maths symbols taken from Adobe's latin character set

<	`<`	>	`>`	_	`\_`	|	`|`
\	`\backslash`	\	`\setminus`	~	`\sim`	|	`\mid`
•	`\bullet`	{	`\lbrace`	}	`\rbrace`		

4 Appendix

4·1 File suffixes

In this document, 'a `foo` file' refers to a file of the type conventionally given a name ending in '`-.foo`' (using lower case because TEX file names are always given in lower case). This table lists some conventional file name suffixes used for files in this package.

Table 5 File suffixes used in this package

Suffix	*Origin*	*Meaning*
`1`	Unix	Manual page for a program, in `nroff` format.
`def`	LaTeX 2e	Definitions used by macro files but not expected to be referred to directly in user documents.
`dtx`	LaTeX 2e	Documented TEX macros—a file which combines macro definitions with their printed documentation.
`eps`	Adobe	An EPSF (Encapsulated PostScript Format) file.
`fig`	Fig	A picture file in Fig's undocumented format.
`fnt`	PDCMAC	Font list—a list of fonts used in a document, generated by the PDCFSEL macros.
`in`	GNU	Template for a configuration file—when using the `configure` script, the file `foo` is generated from the template `foo.in`.
`tex`	TEX	(1) A plain TEX document. (2) A plain TEX definition file.
`tgz`	GNU	A Unix `tar` archive, compressed with GNU `zip`. (Same as `tar.gz`.)
`txt`	traditional	Plain ASCII text, readable on the terminal.
`zip`	?PKZIP	An MS-DOS PKZIP archive.

4·2 List of files

Here is a list of files supplied with the package. A list of the files generated from these—the definition files, used in documents—form the next section.

All the names are chosen so that they may be copied onto, say, an ISO 9660[7] or MS-DOS file system and back to a sensible file system without the names being changed.

Table 6 Files supplied in the package.

File	*Contents*
`00readme.txt`	Brief description of the package.
`configure`	A shell script used to automatically configure the makefile for Unix systems. (This is an unavoidable exception to the rule that names are ISO-9660-compatible.)

[7] See note 1 on page 2.

`copying.tex`	A copy of the GNU General Public Licence, in TEXable form.
`copying.txt`	A copy of the GNU General Public Licence.
`dtxtags`	Shellscript for making tag files in `etags`(1) format.
`dtxtags.1`	A Unix manual entry for `dtxtags`.[8]
`fig2epsf`	A Unix shellscript that converts figures from Fig's format into Encapsulated PostScript Format (EPSF) version 3·0 files. It uses `fig2dev` (from the TransFig package) to do most of the work. (It munges the PostScript code produced by `fig2dev` 2.1.4.1 so that it will print and will work with Ghostview.)
`fig2epsf.1`	A Unix manual page for `fig2epsf`.
`install.txt`	Installation hints.
`makefile.in`	Template from which the configuration process generates a makefile, used by Unix's `make` command to automate compilation and installation. Should be called `Makefile.in` but that's not ISO-9660-compliant.
`magrmac.dtx`	Documentation for Malvern Greek macros. This replaces the file `magrmac.tex` included in Malvern release 1·2.
`magrman.tex`	Brief user manual for `magrmac.tex`. This replaces the version distributed with Malvern 1·2. It requires some Malvern G fonts.
`oput01.eps oput02.eps`	Diagrams for `pdcoput5.dtx`.[9]
`oput01.fig oput02.fig`	Source code for the above figures (Fig format).
`pdcadobe.dtx`	Source code and documentation for `pdcadobe.tex`.
`pdccode.dtx`	Source code and documentation for `pdccode.tex`.
`pdccode.tex`	Macros used by `dtx` files. This file has to be included because `pdccode.dtx` can't be TEXed without it.
`pdccode2.tex`	An experimental variation allowing multiple simultaneous code files.
`pdcfmt2.dtx`	Source code and documentation for formatting macros.
`pdcfsel.dtx`	Source code and documentation for font selection macros.
`pdcguide.dvi`	A copy of the user guide, already run through TEX.
`pdcguide.tex`	This user guide for PDCMAC.
`pdcl1maa.dtx`	Source code and documentation for `pdcl1maa.tex`.
`pdcmacvn.tex`	Version number for the whole package.
`pdcmisc.dtx`	Source code and documentation for some small macro files.
`pdcmondo.tex`	Makes a combined listing of all the `dtx` files (except `pdcsty.dtx`).
`pdcoput5.dtx`	Source code and documentation for an output routine.

[8] This (and three more shellscripts used in the makefile) are not intended to be installed anywhere, but I included manual pages just in case they are—or in case the installer is curious as to what these scripts do.

[9] The second edition of the *PostScript Language Reference Manual* says these should be called '`-.epsf`', but such names are not ISO-9660-compliant, so I have switched to '`-.eps`'.

`pdcsty.dtx`	Source code and documentation for style files (`pdccmdoc.tex`, `ma55doc.tex`, etc.).
`pinstall`	A Unix shellscript that substitutes for the `install` command on systems which don't have GNU `install`.[10]
`pinstall.1`	A Unix manual page for `pinstall`.
`pmkdir`	A Unix shellscript used to create a directory. Unlike plain `mkdir`, it creates parent directories of the specified directory if they do not exist. This would be called `pmkdirhier` but that name is not ISO-9660-compliant.
`pmkdir.1`	A Unix manual page for `pmkdir`.
`version.txt`	List of the version identifiers of the `dtx` files and the shellscripts that come with the package.

4·3 List of definition files

The following files are the ones that are intended to go in the TEX inputs area and to be used in documents. Here a ***macro file*** is simply a file of TEX definitions; a ***style file*** is a higher-level definition file that specifies most of the things that affect the style of a document (layout, fonts, macros, etc.). Style files start by reading a bunch of macro files.

Table 7 Files generated from the `dtx` files.

File	*Contents*
`ma55doc.tex`	Style file for short documents with Malvern 55 as the text font. A table of contents and page headlines are generated automatically.
`ma55lft.tex`	Style file for very short documents with Malvern 55 as the text font. 'Leaflet'-class documents have no table of contents.
`magrmac.tex`	Macros for typesetting in Greek with Malvern fonts (or any font with the Malvern G encoding). There is a brief user's guide in `magrman.tex`.
`pdcadobe.tex`	Support for fonts with the Adobe Standard Roman and Adobe Symbol repertoires[11] in the `dvi` file, and ISO 8859–1 (Latin-1) conventions in the manuscript file. The output encoding actually used is that variation on TEX Text generated by the `afm2tfm` that comes with DVIPS. Newer TEX systems should instead use PostScript fonts with (a subset of) the 1990 Cork encoding.[12] This file actually combines two functions: (1) making the various symbols availabe via commands like `\pounds` and (2) making Latin-1 characters

[10] I am not going to bother trying to make a makefile that will work with all the different versions of `install`, since there is no easy way to tell them apart and they are mutually incompatible.

[11] The ***repertoire*** of an encoding scheme is the set of characters/glyphs it includes. Since PostScript fonts may be easily re-encoded, repertoire is more significant then the actual encoding.

[12] Variously called 'DC', 'EC', 'T1', and 'TEX Extended Text—Latin', and described in *TUGboat* 10#4.

in the manuscript produce corresponding characters in the output. Described in `pdcadobe.dtx`.

`pdccmdoc.tex` Style file for short documents with Computer Modern Roman as the text font. A table of contents and page headlines are generated automatically.

`pdccmlft.tex` Style file for very short documents with Computer Modern Roman as the text font. 'Leaflet'-class documents have no table of contents.

`pdcfmt2.tex` Macros for formatting text—bulleted and numbered lists, syntax descriptions, verbatim text, headings, etc. Most of the facilities used by the style files come from this file.

`pdcfsel.tex` Macros for selecting fonts. Fonts are organized into ***fontset***s (selected with macros with names like `\bodyfonts`) in which fonts are selected with nicknames like `\it`, `\bf` (specified at the start of the document using template macros).

`pdchyex.tex` Some random British English hyphenation exceptions (developed while I was using American English hyphenation paterns). You may not want to use this. Described in `pdcmisc.dtx`.

`pdcimth.tex` Make letters in maths formulas come out in text italic instead of math italic. Useful if the body font isn't CMR, or if multiple-letter identifiers are used. Described in `pdcmisc.dtx`.

`pdcl1maa.tex` Support for documents with using the ISO 8859–1 (Latin-1) character set in the manuscript file and fonts with Malvern A encoding in the `dvi` file. Described in `pdcl1maa.dtx`.

`pdccmsub.tex` Define some Malvern A and PostScript glyphs (like ¥, ®) by overprinting CM glyphs. Described in `pdcmisc.dtx`.

`pdcmigr.tex` Make Greek capitals in maths mode use math italic (fam 1) letters instead of letters from the roman font. Especially useful when there are no Greek caps in the roman font. Described in `pdcmisc.dtx`.

4·4 Obselete files

The following macro files were included with the Malvern 1·0 distribution, but were not intended to be installed. Nevertheless they appear to have been copied into some older versions of the UnixTEX distribution. They are obselete, and their successors have new names (intended to reduce the chance of accidental clashes). I'd appreciate people removing them from their TEX systems.

```
formac.tex   parmac.tex   utils.tex  ssoutput.tex
ldfonts.tex  malvern.tex  cmdoc.tex
```

The following documents are similarly obselete and should not be in the macros directory anyway.

```
aboutmalvern.tex  latexfmv.tex
```

The following macro files have the new-style names, but are superseded by PDCMAC 1·0 files. You are not required to remove then

if you have documents using them. Fortunately, they appear not to have been absorbed by the UnixTEX distribution anyway.

```
pdcfmt.tex  pdcpars.tex  pdcutil.tex  pdcoput.tex
```

The new versions will have the major number of their version ID appended to their names (e.g., `pdcfmt2.tex`), and this way new and old versions may coexist, allowing older documents to still be processed by TEX. The functionality of `pdcpars.tex` and `pdcfmt.tex` have been taken over by `pdcfmt2.tex`.

4·5 GNU General Public License[13]

Version 2, June 1991

Preamble The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software–to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

1. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The 'Program', below, refers to any such program or work, and a 'work based on the Program' means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term 'modification'.) Each licensee is addressed as 'you'.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having

[13] This licence is reproduced verbatim, therefore the American spelling is used.

been made by running the Program). Whether that is true depends on what the Program does.

2. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

3. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

(*a*) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

(*b*) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

(*c*) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

4. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

(*a*) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

(*b*) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

(*c*) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distri-

bution of the source code, even though third parties are not compelled to copy the source along with the object code.

5. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

6. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

7. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

8. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

9. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

10. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and 'any later version', you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

11. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

12. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM 'AS IS'

WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

13. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the 'copyright' line and a pointer to where the full notice is found.

> *one line to give the program's name and a brief idea of what it does*
> Copyright (C) 19*yy name of author*
>
> This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.
>
> This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.
>
> You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

> Gnomovision version 69, Copyright (C) 19YY NAME OF AUTHOR Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items—whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a 'copyright disclaimer' for the program, if necessary. Here is a sample; alter the names:

> Yoyodyne, Inc., hereby disclaims all copyright interest in the program 'Gnomovision' (which makes passes at compilers) written by James Hacker.
>
> SIGNATURE OF TY COON, 1 April 1989
> Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

P D C M A C User Guide

Edition 1 for Release 1·0·01

P. Damian Cugley

Oxford University Computing Laboratory
Parks Road
Oxford OX2 7HN
UK

damian.cugley@comlab.ox.ac.uk

Abstract

PDCMAC is a collection of macro files intended to be useful with TEX formats with similar conventions to those described in the *TEXbook*. This document describes how to unpack the files and use them in TEX documents.

Contents