

The footnote* package

Mark Wooding

28 January 1997

Contents

1	User guide	1	2.4 Hacking existing environ- ments	7
2	Implementation	2		
	2.1 Building footnote text . .	3		
	2.2 Footnote saving	3	A The GNU General Public	
	2.3 The footnote environment	5	License	8

1 User guide

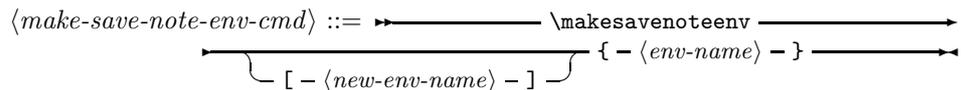
This package provides some commands for handling footnotes slightly better than L^AT_EX usually does; there are several commands and environments (notably `\parbox`, `minipage` and `tabular`¹) which ‘trap’ footnotes so that they can’t escape and appear at the bottom of the page.

`savenotes` The `savenotes` environment saves up any footnotes encountered within it, and performs them all at the end.

`\savenotes` If you’re defining a command or environment, you can use the `\savenotes`
`\spewnotes` command to start saving up footnotes, and the `\spewnotes` command to execute them all at the end. Note that `\savenotes` and `\spewnotes` enclose a group, so watch out. You can safely nest the commands and environments – they work out if they’re already working and behave appropriately.

`minipage*` To help things along a bit, the package provides a `*`-version of the `minipage` environment, which doesn’t trap footnotes for itself (and in fact sends any footnotes it contains to the bottom of the page, where they belong).

`\makesavenoteenv` The new `minipage*` environment was created with a magic command called `\makesavenoteenv`. It has a fairly simple syntax:



*The footnote package is currently at version 1.13, dated 28 January 1997.

¹The `mdwtab` package, provided in this distribution, handles footnotes correctly anyway; it uses an internal version of this package to do so.

Without the optional argument, it redefines the named environment so that it handles footnotes correctly. With the optional argument, it makes the new environment named by $\langle new-env-name \rangle$ into a footnote-friendly version of the $\langle env-name \rangle$ environment.

`\parbox` The package also redefines the `\parbox` command so that it works properly with footnotes.

`footnote` The other problem which people tend to experience with footnotes is that you can't put verbatim text (with the `\verb` command or the `verbatim` environment) into the `\footnote` command's argument. This package provides a `footnote environment`, which *does* allow verbatim things. You use the environment just like you do the command. It's really easy. It even has an optional argument, which works the same way.

`footnotetext` To go with the `footnote` environment, there's a `footnotetext` environment, which just puts the text in the bottom of the page, like `\footnotetext` does.

There's a snag with these environments, though. Some other nonstandard environments, like `tabularx`, try to handle footnotes their own way, because they won't work otherwise. The way they do this is not compatible with the way that the `footnote` and `footnotetext` environments work, and you will get strange results if you try (there'll be odd vertical spacing, and the footnote text may well be incorrect).²

2 Implementation

Most implementations of footnote-saving (in particular, that used in the `tabularx` and `longtable` packages) use a token list register to store the footnote text, and then expand it when whatever was preventing footnotes (usually a `vbox`) stops. This is no good at all if the footnotes contain things which might not be there by the time the expansion occurs. For example, references to things in temporary boxes won't work.

This implementation therefore stores the footnotes up in a box register. This must be just as valid as using tokens, because all I'm going to do at the end is unbox the box).

```
1 \*macro | package
2 \ifx\fn@notes\@undefined%
3   \newbox\fn@notes%
4 \fi
```

I'll need a length to tell me how wide the footnotes should be at the moment.

```
5 \newdimen\fn@width
```

Of course, I can't set this up until I actually start saving footnotes. Until then I'll use `\columnwidth` (which works in `multicol` even though it doesn't have any right to).

```
6 \let\fn@colwidth\columnwidth
```

And now a switch to remember if we're already handling footnotes,

```
7 \newif\if@savingnotes
```

²The solution to this problem is to send mail to David Carlisle persuading him to use this package to handle footnotes, rather than doing it his way.

2.1 Building footnote text

I need to emulate L^AT_EX's footnote handling when I'm putting the notes into my box; this is also useful in the verbatim-in-footnotes stuff.

`\fn@startnote` Here's how a footnote gets started. Most of the code here is stolen from `\@footnotetext`.

```
8 \def\fn@startnote{%
9   \hsize\fn@colwidth%
10  \interlinepenalty\interfootnotelinepenalty%
11  \reset@font\footnotesize%
12  \floatingpenalty\@MM% Is this right???
13  \@parboxrestore%
14  \protected@edef\@currentlabel{\csname p@\@mpfn\endcsname\@thefnmark}%
15  \color@begingroup%
16 }
```

`\fn@endnote` Footnotes are finished off by this macro. This is the easy bit.

```
17 \let\fn@endnote\color@endgroup
```

2.2 Footnote saving

`\fn@fntext` Now to define how to actually do footnotes. I'll just add the notes to the bottom of the footnote box I'm building.

There's some hacking added here to handle the case that a footnote is in an `\intertext` command within a broken `amsmath` alignment environment – otherwise the footnotes get duplicated due to the way that that package measures equations.³

```
18 \long\def\fn@fntext#1{%
19   \ifx\ifmeasuring@\@undefined%
20     \expandafter\@secondoftwo\else\expandafter\@iden%
21     \fi%
22   {\ifmeasuring@\expandafter\@gobble\else\expandafter\@iden\fi}%
23   {%
24     \global\setbox\fn@notes\vbox{%
25       \unvbox\fn@notes%
26       \fn@startnote%
27       \@makefntext{#1%
28         \rule\z@\footnotesep%
29         \ignorespaces%
30         #1%
31         \@finalstrut\strutbox%
32       }%
33       \fn@endnote%
34     }%
35   }%
36 }
```

³The correct solution of course is to implement aligning environments in a sensible way, by building the table and leaving penalties describing the intended format, and then pick that apart in a postprocessing phase. If I get the time, I'll start working on this again. I have a design worked out and the beginnings of an implementation, but it's going to be a long time coming.

`\savenotes` The `\savenotes` declaration starts saving footnotes, to be spewed at a later date. We'll also remember which counter we're meant to use, and redefine the footnotes used by `minipages`.

The idea here is that we'll gather up footnotes within the environment, and output them in whatever format they were being typeset outside the environment.

I'll take this a bit at a time. The start is easy: we need a group in which to keep our local definitions.

```
37 \def\savenotes{%
38   \begingroup%
```

Now, if I'm already saving footnotes away, I won't bother doing anything here. Otherwise I need to start hacking, and set the switch.

```
39   \if@savingnotes\else%
40     \@savingnotestruer%
```

I redefine the `\@footnotetext` command, which is responsible for adding a footnote to the appropriate insert. I'll redefine both the current version, and `minipage`'s specific version, in case there's a nested `minipage`.

```
41   \let\@footnotetext\fn@fntext%
42   \let\@mpfootnotetext\fn@fntext%
```

I'd better make sure my box is empty before I start, and I must set up the column width so that later changes (e.g., in `minipage`) don't upset things too much.

```
43   \fn@width\columnwidth%
44   \let\fn@colwidth\fn@width%
45   \global\setbox\fn@notes\box\voidb@x%
```

Now for some yuckiness. I want to ensure that `minipage` doesn't change how footnotes are handled once I've taken charge. I'll store the current values of `\thempfn` (which typesets a footnote marker) and `\@mpfn` (which contains the name of the current footnote counter).

```
46   \let\fn@thempfn\thempfn%
47   \let\fn@mpfn\@mpfn%
```

The `minipage` environment provides a hook, called `\@minipagerestore`. Initially it's set to `\relax`, which is unfortunately unexpandable, so if I want to add code to it, I must check this possibility. I'll make it `\@empty` (which expands to nothing) if it's still `\relax`. Then I'll add my code to the hook, to override `\thempfn` and `\@mpfn` set up by `minipage`.

Note that I can't just force the `mpfootnote` counter to be equal to the `footnote` one, because `minipage` clears `\c@mpfootnote` to zero when it starts. This method will ensure that even so, the current counter works OK.

```
48   \ifx\@minipagerestore\relax\let\@minipagerestore\@empty\fi%
49   \expandafter\def\expandafter\@minipagerestore\expandafter{%
50     \@minipagerestore%
51     \let\thempfn\fn@thempfn%
52     \let\@mpfn\fn@mpfn%
53   }%
54   \fi%
55 }
```

`\spewnotes` Now I can spew out the notes we saved. This is a bit messy, actually. Since the standard `\@footnotetext` implementation tries to insert funny struts and things, I must be a bit careful. I'll disable all this bits which start paragraphs prematurely.

```
56 \def\spewnotes{%
57   \endgroup%
58   \if@savingnotes\else\ifvoid\fn@notes\else\beginingroup%
59     \let\@makefn@text\@empty%
60     \let\@finalstrut\@gobble%
61     \let\rule\@gobbletwo%
62     \@footnotetext{\unvbox\fn@notes}%
63   \endgroup\fi\fi%
64 }
```

Now make an environment, for users.

```
65 \let\endsavenotes\spewnotes
```

That's all that needs to be in the shared code section.

```
66 </macro | package>
67 <*package>
```

2.3 The footnote environment

Since `\footnote` is a command with an argument, things like `verbatim` are unwelcome in it. Every so often someone on `comp.text.tex` moans about it and I post a nasty hack to make it work. However, as a more permanent and 'official' solution, here's an environment which does the job rather better. Lots of this is based on code from my latest attempt on the newsgroup.

I'll work on this in a funny order, although I think it's easier to understand. First, I'll do some macros for reading the optional argument of footnote-related commands.

`\fn@getmark` Saying `\fn@getmark{<default-code>}{<cont-code>}` will read an optional argument giving a value for the footnote counter; if the argument isn't there, the `<default-code>` is executed, and it's expected to set up the appropriate counter to the current value. The footnote marker text is stored in the macro `\@thefnmark`, as is conventional for L^AT_EX's footnote handling macros. Once this is done properly, the `<cont-code>` is called to continue handling things.

Since the handling of the optional argument plays with the footnote counter locally, I'll start a group right now to save some code. Then I'll decide what to do based on the presence of the argument.

```
68 \def\fn@getmark#1#2{%
69   \beginingroup%
70   \@ifnextchar[%
71     {\fn@getmark@i{#1}}%
72     {#1\fn@getmark@ii{#2}}%
73 }
```

There's an optional argument, so I need to read it and assign it to the footnote counter.

```
74 \def\fn@getmark@i#1[#2]{%
75   \csname c@\mpfn\endcsname#2%
```

```
76 \fn@getmark@ii%
77 }
```

Finally, set up the macro properly, and end the group.

```
78 \def\fn@getmark@ii#1{%
79 \unrestored@protected@xdef\@thefnmark{\thempfn}%
80 \endgroup%
81 #1%
82 }
```

From argument reading, I'll move on to footnote typesetting.

`\fn@startfntext` The `\fn@startfntext` macro sets everything up for building the footnote in a box register, ready for unboxing into the footnotes insert. The `\fn@prefntext` macro is a style hook I'll set up later.

```
83 \def\fn@startfntext{%
84 \setbox\z@\vbox\bgroup%
85 \fn@startnote%
86 \fn@prefntext%
87 \rule\z@\footnotesep%
88 \ignorespaces%
89 }
```

`\fn@endfntext` Now I'll end the vbox, and add it to the footnote insertion. Again, I must be careful to prevent `\@footnotetext` from adding horizontal mode things in bad places.

```
90 \def\fn@endfntext{%
91 \@finalstrut\strutbox%
92 \fn@postfntext%
93 \egroup%
94 \beginingroup%
95 \let\@makefntext\@empty%
96 \let\@finalstrut\@gobble%
97 \let\rule\@gobbletwo%
98 \@footnotetext{\unvbox\z}%
99 \endgroup%
100 }
```

`footnote` I can now start on the environment proper. First I'll look for an optional argument.

```
\def\footnote{%
```

Oh. I've already come up against the first problem: that name's already used. I'd better save the original version.

```
101 \let\fn@latex@@footnote\footnote
```

The best way I can think of for seeing if I'm in an environment is to look at `\@currentenv`. I'll need something to compare with, then.

```
102 \def\fn@footnote{footnote}
```

Now to start properly. ;-)

```
103 \def\footnote{%
104 \ifx\@currentenv\fn@footnote%
```

```

105   \expandafter\@firstoftwo%
106   \else%
107   \expandafter\@secondoftwo%
108   \fi%
109   {\fn@getmark{\stepcounter\@mpfn}%
110    {\leavevmode\unskip\@footnotemark\fn@startfntext}}%
111   {\fn@latex@@footnote}%
112 }

```

Ending the environment is simple.

```
113 \let\endfootnote\fn@endfntext
```

`footnotetext` I'll do the same magic as before for `\footnotetext`.

```

114 \def\fn@footnotetext{footnotetext}
115 \let\fn@latex@@footnotetext\footnotetext
116 \def\footnotetext{%
117   \ifx\@currenvir\fn@footnotetext%
118     \expandafter\@firstoftwo%
119   \else%
120     \expandafter\@secondoftwo%
121   \fi%
122   {\fn@getmark{}\fn@startfntext}%
123   {\fn@latex@@footnotetext}%
124 }
125 \let\endfootnotetext\endfootnote

```

`\fn@prefntext` Now for one final problem. The style hook for footnotes is the command
`\fn@postfntext` `\@makefntext`, which takes the footnote text as its argument. Clearly this is utterly unsuitable, so I need to split it into two bits, where the argument is. This is very tricky, and doesn't deserve to work, although it appears to be a good deal more effective than it has any right to be.

```

126 \long\def\@tempa#1\@@#2\@@@{\def\fn@prefntext{#1}\def\fn@postfntext{#2}}
127 \expandafter\@tempa\@makefntext\@@\@@@

```

2.4 Hacking existing environments

Some existing L^AT_EX environments ought to have footnote handling but don't. Now's our chance.

`\makesavenoteenv` The `\makesavenoteenv` command makes an environment save footnotes around itself.

It would also be nice to make `\parbox` work with footnotes. I'll do this later.

```
128 \def\makesavenoteenv{\@ifnextchar[\fn@msne@ii\fn@msne@i}
```

We're meant to redefine the environment. We'll copy it (using `\let`) to a magic name, and then pass it on to stage 2.

```

129 \def\fn@msne@i#1{%
130   \expandafter\let\csname msne$#1\expandafter\endcsname%
131   \csname #1\endcsname%
132   \expandafter\let\csname endmsne$#1\expandafter\endcsname%
133   \csname end#1\endcsname%
134   \fn@msne@ii[#1]{msne$#1}%
135 }

```

Now we'll define the new environment. The start is really easy, since we just need to insert a `\savenotes`. The end is more complex, since we need to preserve the `\if@endpe` flag so that `\end` can pick it up. I reckon that proper hooks should be added to `\begin` and `\end` so that environments can define things to be done outside the main group as well as within it; still, we can't all have what we want, can we?

```

136 \def\fn@msne@ii[#1]#2{%
137   \expandafter\edef\csname#1\endcsname{%
138     \noexpand\savenotes%
139     \expandafter\noexpand\csname#2\endcsname%
140   }%
141   \expandafter\edef\csname end#1\endcsname{%
142     \expandafter\noexpand\csname end#2\endcsname%
143     \noexpand\expandafter%
144     \noexpand\spewnotes%
145     \noexpand\if@endpe\noexpand\@endpetrue\noexpand\fi%
146   }%
147 }

```

`minipage*` Let's define a `minipage*` environment which handles footnotes nicely. Really easy:

```

148 \makesavenoteenv[minipage*]{minipage}

```

`\parbox` Now to alter `\parbox` slightly, so that it handles footnotes properly. I'm going to do this fairly inefficiently, because I'm going to try and change it as little as possible.

First, I'll save the old `\parbox` command. If I don't find a '*', I'll just call this command.

```

149 \let\fn@parbox\parbox

```

This is the clever bit: I don't know how many optional arguments Mr Mittelbach and his chums will add to `\parbox`, so I'll handle any number. I'll store them all up in my first argument and call myself every time I find a new one. If I run out of optional arguments, I'll call the original `\parbox` command, surrounding it with `\savenotes` and `\spewnotes`.

```

150 \def\parbox{\ifnextchar[{\fn@parbox@i}{\fn@parbox@ii}}
151 \def\fn@parbox@i#1[#2]{%
152   \ifnextchar[{\fn@parbox@i{#1[#2]}}{\fn@parbox@ii{#1[#2]}}%
153 }
154 \long\def\fn@parbox@ii#1#2#3{\savenotes\fn@parbox#1{#2}{#3}\spewnotes}

```

Done!

```

155 </package>

```

Mark Wooding, 28 January 1997

Appendix

A The GNU General Public License

The following is the text of the GNU General Public License, under the terms of which this software is distributed.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright © 1989, 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

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

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

1. 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.

2. 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.

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

4. 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.
5. 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.
6. 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.
7. 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.

8. 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.

9. 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.

10. 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

11. 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.
12. 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

Appendix: 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) yyyy 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., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, 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) yyyy 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.

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

Symbols	F
<code>\@@</code> 126, 127	<code>\floatingpenalty</code> 12
<code>\@@@</code> 126, 127	<code>\fn@colwidth</code> 6, 9, 44
<code>\@@undefined</code> 2, 19	<code>\fn@endfntext</code> <u>90</u> , 113
<code>\@MM</code> 12	<code>\fn@endnote</code> <u>17</u> , 33
<code>\@currentlabel</code> 14	<code>\fn@fntext</code> <u>18</u> , 41, 42
<code>\@currenenvir</code> 104, 117	<code>\fn@footnote</code> 102, 104
<code>\@endpetrue</code> 145	<code>\fn@footnotetext</code> 114, 117
<code>\@finalstrut</code> 31, 60, 91, 96	<code>\fn@getmark</code> <u>68</u> , 109, 122
<code>\@firstoftwo</code> 105, 118	<code>\fn@getmark@i</code> 71, 74
<code>\@footnotemark</code> 110	<code>\fn@getmark@ii</code> 72, 76, 78
<code>\@footnotetext</code> 41, 62, 98	<code>\fn@latex@footnote</code> 101, 111
<code>\@gobble</code> 22, 60, 96	<code>\fn@latex@footnotetext</code> 115, 123
<code>\@gobbletwo</code> 61, 97	<code>\fn@mpfn</code> 47, 52
<code>\@iden</code> 20, 22	<code>\fn@msne@i</code> 128, 129
<code>\@ifnextchar</code> 70, 128, 150, 152	<code>\fn@msne@ii</code> 128, 134, 136
<code>\@makefntext</code> 27, 59, 95, 127	<code>\fn@notes</code> 2, 3, 24, 25, 45, 58, 62
<code>\@minipagerestore</code> 48–50	<code>\fn@parbox</code> 149, 154
<code>\@mpfn</code> 14, 47, 52, 75, 109	<code>\fn@parbox@i</code> 150–152
<code>\@mpfootnotetext</code> 42	<code>\fn@parbox@ii</code> 150, 152, 154
<code>\@parboxrestore</code> 13	<code>\fn@postfntext</code> 92, <u>126</u>
<code>\@savingnotestrue</code> 40	<code>\fn@prefntext</code> 86, <u>126</u>
<code>\@secondoftwo</code> 20, 107, 120	<code>\fn@startfntext</code> <u>83</u> , 110, 122
<code>\@tempa</code> 126, 127	<code>\fn@startnote</code> <u>8</u> , 26, 85
<code>\@thefnmark</code> 14, 79	<code>\fn@thempfn</code> 46, 51
	<code>\fn@width</code> 5, 43, 44
	<code>\footnote</code> 101, 103
	<code>footnote</code> (environment) 2, <u>101</u>
	<code>\footnotesep</code> 28, 87
	<code>\footnotesize</code> 11
	<code>\footnotetext</code> 115, 116
	<code>footnotetext</code> (environment) 2, <u>114</u>
	I
	<code>\if@endpe</code> 145
	<code>\if@savingnotes</code> 7, 39, 58
	<code>\ifmeasuring@</code> 19, 22
	<code>\ignorespaces</code> 29, 88
	<code>\interfootnotelinepenalty</code> 10
	<code>\interlinepenalty</code> 10
C	
<code>\color@begingroup</code> 15	
<code>\color@endgroup</code> 17	
<code>\columnwidth</code> 6, 43	
E	
<code>\endfootnote</code> 113, 125	
<code>\endfootnotetext</code> 125	
<code>\endsavenotes</code> 65	
environments:	
<code>footnote</code> 2, <u>101</u>	
<code>footnotetext</code> 2, <u>114</u>	
<code>minipage*</code> 1, <u>148</u>	
<code>savenotes</code> 1	

	L			
\leavevmode	110	\rule 28, 61, 87, 97
	M		S	
\makesavenoteenv	1, <u>128</u> , 148	\savenotes 1, <u>37</u> , 138, 154
minipage* (environment)	1, <u>148</u>	savenotes (environment) 1
	N		\spewnotes 1, <u>56</u> , 65, 144, 154
\newbox	3	\stepcounter 109
\newdimen	5	\strutbox 31, 91
\newif	7		T
	P		\thempfn 46, 51, 79
\parbox	2, <u>149</u>		U
\protected@edef	14	\unrestored@protected@xdef 79
	R			V
\reset@font	11	\voidb@x 45