GNU Talk Filters

Amusing Text Translators
Version 2.3
8 December 2003

Mark A. Lindner, et. al.

Table of Contents

1 Introduction

The GNU Talk Filters are filter programs that convert English text into text that mimics a stereotyped or otherwise humorous type of speech. This package is not an original work but rather a collection and integration of existing filter programs that were written by various people and that have been in the public domain for many years. For a list of authors, see Chapter 5 [Credits], page 13.

The filters are provided in both executable and library form. See Chapter 2 [The Filters], page 3 for synopses of the programs, and Chapter 3 [The Talkfilters Library], page 5 for a description of the library API.

Since the filters do word and substring substitution on the text they process, any word-wrap formatting of the original text will not be preserved in the output. The included `wrap` filter program may be used to reformat the output of the other filters to fit within a specified number of columns.

Some of these filters contain vulgarity, and thus are not appropriate for all audiences. If you find something offensive in one or more of these filters, please do not flame the maintainer of this package or the original authors, or request that the filter(s) in question be censored or removed; requests of this type will be summarily ignored.

These filters are not guaranteed to be idempotent across all inputs; that is, repeated applications of a given filter on an input may cause the output to differ each time. Moreover, some of the filters use randomization techniques so a given input is not guaranteed to produce the same output across invocations.

The ‘flex’ lexer (or any other lexer program, for that matter) is not required to build and use this package. However, ‘flex’ *is* required to rebuild the filters if the ‘`.l`’ source files are changed.

These filters are provided for amusement only. No racial or societal slurs are intended nor should be inferred.

2 The Filters

The filters provided in this package are enumerated below. This chapter describes the synopses for the individual filter programs; for a description of the library API, see Chapter 3 [The Talkfilters Library], page 5.

`austro`	Austrian (Ahhhhnold)
`b1ff`	B1FF of Usenet yore
`brooklyn`	Brooklyn accent
`chef`	Swedish Chef (from *The Muppet Show*)
`cockney`	Londoner accent
`drawl`	Southern drawl
`dubya`	George "Dubya" Bush
`fudd`	Elmer Fudd (from the Looney Tunes cartoons)
`funetak`	Thick Asian accent
`jethro`	Jethro from *The Beverly Hillbillies*
`jive`	1970's Jive
`kraut`	German accent
`pansy`	Effeminate male
`pirate`	Pirate talk
`postmodern`	Postmodernist talk ("Feminazi")
`redneck`	Country redneck
`valspeak`	Valley talk
`warez`	H4x0r code
`wrap`	Word-wrap filter

The filter programs read from standard input and write to standard output. They all recognize the following switches:

`--version`
Print version information and exit.

`--help` Print usage information and exit.

The `wrap` filter program additionally recognizes the following switch:

`-w` ***`width`*** Specify the maximum number of columns *width* that the text may span. The minimum value of *width* is 10.

An example usage might be:

```
man ls | jive | wrap -w 78 | less
```

3 The Talkfilters Library

All of the filters in the GNU Talk Filters are available collectively as a C library which can be linked with other programs to provide embedded text filtering support. While the individual filter programs filter from standard input to standard output, the filtering functions in the library operate on in-memory buffers instead.

This chapter describes the API to the Talkfilters library. All of the functions and types described below are declared in the header file '`talkfilters.h`'.

`int gtf_filter_count` (*void*) [Function]

This function returns the number of filters in the library.

`gtf_filter_t * gtf_filter_list` (*void*) [Function]

This function returns a pointer to an array of *gtf_filter_t* structures which contain information about each of the filters in the library, including the filter's symbolic name, a brief description of the filter that is suitable for display purposes, and a pointer to the filter function.

The following C program illustrates the use of `gtf_filter_count()` and `gtf_filter_list()` to display information about each filter in the library and invoke the filter on some test input:

```
#include <stdio.h>
#include <talkfilters.h>

int main(void)
  {
  int ct, i;
  const gtf_filter_t *filters, *fp;
  const char *inbuf = "This is a test.";
  char outbuf[1024];

  ct = gtf_filter_count();
  printf("There are %d filters available.\n", ct);
  filters = gtf_filter_list();
  for(i = 0, fp = filters; i < ct; i++, fp++)
    {
    printf("filter #%d: %s - %s\n", i + 1, fp->name, fp->desc);
    fp->filter(inbuf, outbuf, sizeof(outbuf));
    puts(outbuf);
    }

  exit(0);
  }
```

The type *gtf_filter_t* is a structure which contains the following members:

`char *name`
: The symbolic name of the filter.

`char *desc`
: A brief description of the filter.

`int (*filter)(const char *, char *, size_t)`
: The filter function.

`gtf_filter_t * gtf_filter_lookup` (*const char *`name`*) [Function]

This is a lookup function for locating a specific filter. The function searches for the filter with the symbolic name *name*, and returns a pointer to the *gtf_filter_t* structure for that filter. If a filter with the given name is not found, the function returns `NULL`.

`int gtf_filter_austro` (*const char *`input`, char *`buf`, size_t `bufsz`*) [Function]
`int gtf_filter_b1ff` (*const char *`input`, char *`buf`, size_t `bufsz`*) [Function]
`int gtf_filter_brooklyn` (*const char *`input`, char *`buf`, size_t `bufsz`*) [Function]
`int gtf_filter_chef` (*const char *`input`, char *`buf`, size_t `bufsz`*) [Function]
`int gtf_filter_cockney` (*const char *`input`, char *`buf`, size_t `bufsz`*) [Function]
`int gtf_filter_drawl` (*const char *`input`, char *`buf`, size_t `bufsz`*) [Function]
`int gtf_filter_dubya` (*const char *`input`, char *`buf`, size_t `bufsz`*) [Function]
`int gtf_filter_fudd` (*const char *`input`, char *`buf`, size_t `bufsz`*) [Function]
`int gtf_filter_funetak` (*const char *`input`, char *`buf`, size_t `bufsz`*) [Function]
`int gtf_filter_jethro` (*const char *`input`, char *`buf`, size_t `bufsz`*) [Function]
`int gtf_filter_jive` (*const char *`input`, char *`buf`, size_t `bufsz`*) [Function]
`int gtf_filter_kraut` (*const char *`input`, char *`buf`, size_t `bufsz`*) [Function]
`int gtf_filter_pansy` (*const char *`input`, char *`buf`, size_t `bufsz`*) [Function]
`int gtf_filter_pirate` (*const char *`input`, char *`buf`, size_t `bufsz`*) [Function]
`int gtf_filter_postmodern` (*const char *`input`, char *`buf`, size_t `bufsz`*) [Function]
`int gtf_filter_redneck` (*const char *`input`, char *`buf`, size_t `bufsz`*) [Function]
`int gtf_filter_valspeak` (*const char *`input`, char *`buf`, size_t `bufsz`*) [Function]
`int gtf_filter_warez` (*const char *`input`, char *`buf`, size_t `bufsz`*) [Function]

These functions invoke the corresponding filters on the input buffer *input*, which must be a NUL-terminated string. At most *bufsz* - 1 bytes of output are written to the buffer *buf*. The output is unconditionally NUL-terminated, but the text itself may be truncated if the buffer is too small to accommodate all of the output.

The functions return 0 on success, or 1 if a buffer overflow occurred (signifying that the output was truncated).

Note that lexical scanners generated by 'flex' are not reentrant, so no assumptions should be made about the reentrancy of the above functions. When this library is used in a multithreaded environment, calls to these functions should be protected by mutex locks.

4 Writing New Filters

Writing new filters and adding them to the library is fairly straightforward, but certain conventions must be observed to ensure that the filters will work properly both as standalone programs and as library functions. The internal header file '`common.h`' declares some utility functions and macros that should be used to ensure proper behavior.

`gtf_parse_args` () [Macro]

A filter's `main()` function should make a call to this macro to process the command line arguments. Currently, only the standard '`--help`' and '`--version`' switches are recognized. A call to this macro should typically be the first statement in `main()`.

`gtf_random_seed` () [Macro]

Filters which make calls to the `gtf_random()` macro (described below), should make a call to this macro (preferably in `main()`) to seed the random number generator. This macro seeds the random number sequence with a bitwise OR of the current system time and the PID of the calling process.

`gtf_random` (*range*) [Macro]

This macro returns a random integer in the range [0, *range*).

`gtf_printf` (*format, ...*) [Macro]
`gtf_putc` (*char*) [Macro]
`gtf_puts` (*string*) [Macro]

Filters must be able to function both as standalone programs and as library functions, so the stdio library functions cannot be used to write output. Instead, these macros should be used in place of the stdio library functions `printf()`, `putchar()`, and `puts()`, respectively. When a filter is compiled to run as an executable, these macros simply evaluate to calls to the stdio functions they replace; when it is compiled into the library, they evaluate to calls to internal library functions which write to a data buffer.

`gtf_puts_case` (*string*) [Macro]

This macro is a specialized form of `gtf_puts()` which ensures that the case of the first character in *string* matches that of the first character in the currently matched token. For example, if `yytext` is "Hello", calling `gtf_puts_case()` with either "howdy" or "Howdy" as an argument will write the string "Howdy", whereas if `yytext` is "hello", the string written will be "howdy".

`gtf_unput_last` () [Macro]

This macro "unputs" the last character of the current token. In other words, the last character of `yytext` will be returned back to the input stream, so that it will be the next character read by the lexical scanner.

`void gtf_strbuf_init` (*gtf_databuf_t* ***sbuf***, *char* ***buf***, *size_t* **`bufsz`**) [Function]

This function initializes the *gtf_databuf_t* structure at *sbuf* to point to the buffer *buf*, which is *bufsz* bytes in length; these values specify the buffer to which the `gtf_printf()`, `gtf_putc()`, `gtf_puts()`, and `gtf_puts_case()` macros will ultimately write their output when the filter is called through the library API. The *gtf_databuf_t* structure contains an integer field named `overflow` which will contain the value 1 after the call to `yylex()` if a buffer overflow occurred during filtering; otherwise it will contain the value 0.

`gtf_reset` () [Macro]

This macro should be called after the call to `yylex()` within the filter API function in order to reset the state of the lexical scanner in preparation for the next call.

The following example shows the C code that implements both the library interface and the `main()` function for the `chef` filter.

```
#ifdef LIBRARY_MODE

int gtf_filter_chef(const char *input, char *buf, size_t bufsz)
  {
  gtf_databuf_t buffer;
  YY_BUFFER_STATE _yybuf;

  gtf_strbuf_init(&buffer, buf, bufsz);
  _yybuf = yy_scan_string(input);
  yylex(&buffer);
  yy_delete_buffer(_yybuf);
  gtf_reset();

  return(buffer.overflow);
  }

#else /* LIBRARY_MODE */

int main(int argc, char **argv)
  {
  gtf_parse_args();

  yylex(NULL);

  exit(EXIT_SUCCESS);
  }

#endif /* LIBRARY_MODE */
```

Each filter lex file must have an introductory fragment similar to the following:

```
%option prefix="chef_yy"
%option outfile="lex.yy.c"
%option noyywrap

%{

#include "common.h"

#define YY_DECL int yylex(gtf_databuf_t *buf)

%}
```

The `prefix` option specifies a prefix for the names of the functions generated for this filter; the function names for each filter must be unique so that multiple filters can coexist within the library.

The `outfile` option reverses an undesirable side effect of the `prefix` option, which is to name the generated C source file based on the prefix; this breaks `ylwrap` (an Automake helper program), which expects the output file to be named `lex.yy.c`.

The `noyywrap` option specifies that no `yywrap()` function is needed.

The macro `YY_DECL` is defined to specify that the `yylex()` function takes a single argument, a pointer to the *gtf_databuf_t* structure described above. When `yylex()` is called from `main()`, no output buffer is needed since text is written to standard output, so in that case, it is called with a `NULL` pointer as the argument. This function can be declared to accept additional arguments, but the *buf* argument *must* be present.

The file '`talkfilters.c`' contains a filter registry in the form of an array of structures. Entries should be added therein for new filters. Appropriate `extern` declarations of the API functions for new filters should also be added to '`talkfilters.h`'.

5 Credits

While all of these filters have been available in one form or another in the public domain for many years, the original authors of some of the filters are unknown. Reasonable attempts were made to find the authors and obtain written permission to repackage the filters as GNU software, but in some cases they could not be located.

The following table lists the known authors and contributors.

`austro`	Tom van Nes
`b1ff`	Matt Welsh, David Whitten
`brooklyn`	Daniel V Klein (‘`nyc.l`’)
`chef`	John Hagerman
`cockney`	Stephen K Mulrine, Edward Betts (‘`ken.l`’); *unknown* (‘`cockney.l`’); extensive enhancements by Samuel Stoddard
`drawl`	Adam Hudd
`dubya`	*anonymous contribution*
`fudd`	*unknown*
`funetak`	Eclipse Enterprises
`jethro`	Duane Paulson
`jive`	Daniel V Klein, Clement Cole, with enhancements by Samuel Stoddard
`kraut`	*unknown*
`pansy`	*unknown*
`pirate`	Original Perl/PHP version by Dougal Campbell, with enhancements by Mark Lindner
`postmodern`	*unknown*
`redneck`	Brand Hilton
`valspeak`	*unknown*
`warez`	Ian Johnston, with enhancements by Mark Lindner
`wrap`	Mark Lindner

The filters were repackaged, integrated, optimized, and documented by Mark Lindner (`markl@gnu.org`).

6 Software License

GNU GENERAL PUBLIC LICENSE
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.

GNU GENERAL PUBLIC LICENSE
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., 59 Temple Place - Suite 330, Boston, MA
02111-1307, 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.
```

Type and Function Index