Stanford Exploration Project, Report 72, December 18, 1997, pages 1–18

Electronic document preface

Jon Claerbout[1]

keywords: *algorithm, computing, interactive, work stations*

PREFACE

A technical document rarely has a *preface* because a preface describes the document itself without describing the content. This document partly describes the accompanying electronic document which includes my recent textbook, SEP progress reports 70 and 72, and Joe Dellinger's thesis, SEP report 69.

ABSTRACT

I have created a textbook on echo soundings that provides a model for all publications where the results are contained in figures and illustrations. In this arrangement, programs are linked with the text file. Located in each figure caption is a pushbutton that launches the commands that recreate the figure. This gives new meaning to the hallowed words, "reproducible research." The software is all free and nearly all of it is in widespread use. Experience at the Stanford Exploration Project shows that preparing such documents is little extra effort for those already using LaTeX and makefiles. However, the required software is many megabytes and installing software on UNIX systems is daunting for amateurs. Therefore, I plan to distribute a CD-ROM with software preinstalled for machines of several manufacturers. This gives instant access to my book and it also provides a model of reproducible research. With luck, this document may appear on CD-ROM simultaneously with the paper report.

INTRODUCTION

Macintosh Hypercard is an impressive way of organizing information. Mathematica Journal is an impressive way of preparing and presenting mathematics. Mathematica Journal seems to have some of the ingredients to meet my goals, but I cannot distribute Mathematica freely to readers and users.

Here I describe free software that supplements the free software that many people

[1]**email:** not available

already use. Together this gives new meaning to the hallowed words, “reproducible research”.

The goals of this project are:

- Teach people how to merge a publication with its underlying computational analysis so the work can be verified and extended by other people using other hardware.
- Distribute my new electronic textbook.

To meet these goals I delivered my book to a traditional publisher (Blackwell Scientific Publications) on September 10, 1991. I retained the electronic rights and I grant to the public a GNU Public License to the book software. To try to overcome human resistance to installing software we plan to install binary versions of the most important code for three popular UNIX workstations. The deadline for this paper report is October 18, and the CD-ROM deadline is about November 1. All should be simultaneously available at the November 10 SEG meeting.

OUR ELECTRONIC DOCUMENTS

First, here is a brief review of electronic document experience at the Stanford Exploration Project.

Discovery of the essence of reproducibility

It is a big chore for one researcher to reproduce the analysis and computational results of another. I always thought the difficulty in reproducing other people’s work had deep and human causes. Then one day, quite by accident, I discovered that this problem has a simple technological solution: illustrations (figures) in a technical document are made by programs and command scripts that along with required data should be linked with the document itself. The text file of the document should, for each figure or illustration, contain a pointer to the command script that made it. This is hardly any extra work for the author, but it makes the document much more valuable to readers who possess the document in electronic form because they are able to track down the computations that lead to the illustrations. They can change parameters, data, boundaries, subroutines, etc., much easier than they could starting from scratch with the paper publication.

Verification of reproducibility

A judgement of the reproducibility of research no longer requires an expert—a clerk can do it. To verify reproducibility of research, four steps are required:

1. Issue the "burn illustrations" command.
2. Examine the directories to verify that all files of the form NAME.ps are gone.
3. Issue the "build illustrations" command and wait for its completion.
4. Print the document and verify the presence of the original illustrations.

Advertising the reproducibility

In figure captions, the [R] means we can run the `burn` command to blow away all the figures and a subsequent `cake figures` will rebuild them. When a figure cannot be rebuilt, then the figure name is not `NAME.ps` but `NAME.ps.save` so it is spared by the burn command and the text processor flags the caption with [NR] denoting Non Reproducible. In practice people now use about 30 minutes of compute time for the the cutoff.

There remains an embarrassing bug because we have not had time to build some important software. We have an intermediate plot language called `vplot`. Intermediate plot files can be saved as `NAME.v` (if reproducible) or as `NAME.v.save` (if not). When a `NAME.ps` file is created from a `NAME.v.save` file, the figure has been reproduced in a disappointingly trivial sense. This problem can be attacked in both cake language and TeX language and we will have the most robust result if we do it both ways. Obviously people will always be able to cheat the [R] [NR] convention, but we don't want people to do so inadvertently.

Unfortunately, this bug was not fixed in time for my paper book so that book nowhere defines the meaning of [R] and [NR]. I hope to fix it in time for the first CD-ROM distribution.

Familiar software tools

My document is written in the popular LaTeX word processing language which is freely available. I also use other freely available software that we have merged with our scientific software to produce reports which we preview on a screen. From a software point of view my new electronic book is almost identical to theses and reports prepared in our department, and it is little different from those prepared in many other departments here at Stanford University as well as at many other universities. The book contains figures resulting from mathematical analysis of data. These figures can be recalculated by pressing a pushbutton in the figure caption. I think it is a tragedy that doctoral dissertations do not offer the easy reproducibility that my book does, because it would only be a tiny bit of extra work for their authors. I believe many other people will recognize this too as soon they see how easy it is to do.

We[2] built a figure inclusion command called `activeplot` in LaTeX language which points to a figure-building command script (cakefile). The word processor now requires each author for each figure to supply a directory path and a "target" name. (or else to explicitly state that the figure is irreproducible. Nonreproducible figure file names are of the form NAME.ps.save). While the manuscript is being written, naming the command scripts is only a tiny burden on the author. Experience shows that this bit of information is often the key that gets lost.

Positive feedback

The `activeplot` plot inclusion method worked so well that with one or two weeks effort, Martin Karrenbach was able to convert to the required form a 400 page project report written by 16 people. As a test, we deleted all the figures in the report and issued the rebuild command. Much to our delight, more than three quarters of the figures were properly rebuilt. Perhaps all would have, but some figures were given to us in paper form by outsiders and some were done on outside computers with different software and a few authors were not completely maintaining their figure building scripts.

Unfamiliar software

A bit of software familiar to most people who use UNIX is `make`. We find a less familiar alternative called `cake` is simpler, more powerful, vendor-independent, and free.

What should be on this disk

Our first disk should contain:

- my interactive book, Earth Soundings Analysis, Processing Versus Inversion
- SEP–72
- SEP–70
- Joe Dellinger's thesis

Several items will not make this production deadline, but are prospects for later disks.

- John Etgen's thesis
- Carlos's /q2/movie document

[2]Martin Karrenbach, Steve Cole, and I

- We have about 50 old SEP reports on tape. Many of these are in LaTeX and the others could be converted with Kamal Al Yahya's `tr2tex` program. Perhaps with a few `sed` scripts and the like we could bring it all up. A temporary employee could be hired to make some required manual corrections. It would be nice to integrate Rick Ottolini's index with the documents themselves. The CD-ROM has plenty of space for DVI files for everything.

WHAT IS CD-ROM?

Most of us in scientific computing think of CD-ROM as a disc that holds music and costs about ten dollars. So here I'll explain how CD-ROM can be used as a computer disk.

How does CD-ROM compare to magnetic tape?

We could produce our interactive publications on magnetic tape instead of CD-ROM. Reasons to publish on magnetic tape are: More people have magnetic tape drives. Tape technology is better understood by more people. We can easily make magnetic tapes cheaply and in moderate numbers. A shirt-pocket size magnetic tape will hold 2 gigabytes. That is four times as big as a CD-ROM.

Reasons to produce our interactive publications on CD-ROM are: When we hand a CD-ROM to someone and they put it in their drive, they have instantaneously mounted 680 Mb (megabytes) of our file system in their computer. They did not first need to clear 680 Mb of space in their computer and they did not need to wait two hours for the tape to read. CD-ROM is more suited than magnetic tape to please the recipient. Thus CD-ROM is a better publication medium.

CD-ROM readers are likely to become *more* numerous than tape readers because manufacturers have begun distributing software on CD-ROM and the drives themselves are cheaper than tape readers. We recently purchased some for $ 431 each (including academic discount) whereas 2 Gb tapes cost multiple thousands of dollars.

File systems

A CD-ROM disk is a read-only file. The entire disk holds about 680 Megabytes which could be broken into several tracks but for most software purposes it would be handled as a single track. A UNIX file system is a file with a special form. Such a file can be "mounted" by the UNIX mount command and then it is accessible as a file system (you can snoop around on it using `cd` and `ls` commands). The UNIX `tar` command will take a UNIX directory tree and from it create a single file, but this file will not be in the proper form for a CD-ROM drive. For the material on a CD-ROM to be a file system that can be mounted, the file must be prepared by a tar-like program. One such program that we purchased (with a huge academic discount) is called "makedisk" from a company called Young Minds. Before that we mimicked a CD-ROM with the UNIX mount command, by mounting a file system in read-only mode. I believe you can circumvent the purchase of the Young Minds software by having your `tar` tape formatted (for a price) by the disk manufacturer.

How to overcome the read-only limitation

In our research we are not accustomed to read-only file systems. So we don't know how crippled they are compared to ordinary read-write file systems. Our worst fear was that the software that we put on a CD-ROM will be of no use until someone copies the whole thing into their hard disk. This would make CD-ROM as inconvenient as tape. So now I will explain a well known technique that makes everything easy:

In my back yard is a tree with about 3500 leaves on it and in my computer file system there are about 3500 files. In winter time all the leaves fall off the tree and then light passes directly through it because the leaves have much more surface area than the branches. Likewise the computer files take much more space than the directory structure. (Directories are sometimes called "folders"). So the trick to using a CD-ROM as if it were a read-write file system is to first make a copy of the CD-ROMs directory structure on your hard disk. Like the tree branches, that isn't too voluminous. In this directory structure you do not put files, but pointers (symbolic links) to the files on the CD-ROM. Now whenever it happens that you need to change a file, what you do is replace the link to the CD-ROM by a copy of the file on your hard disk.

The tree without its leaves is often called a "shadow tree". To make one we could use a well-known simple shell program called `lndir` (distributed with X). Instead we use a program called `cd_link` that is free from Young Minds. It works much like `lndir` except that it also does the required name translation from ISO 9660 names to our directory names.

The only fly in the ointment is that it requires about 15-30 minutes to make the shadow tree. Luckily you pay this price only the first time you load the disk.

The ISO 9660 Standard

From the Young Minds documentation we learn that the CD-ROM standard, ISO 9660 requires that file names be restricted in several ways. Basically, 1. have no lower case and 2. fewer than 8 or 11 chars. Makedisk maps our file names into ISO 9660 compliant names. Then after the user mounts his disk, the user needs to run a program 'cd_link' (supplied by YM as C source) that appears to build (on the user's hard disk) a shadow tree with our names and symbolic links to the ISO 9660 named files on the CD-ROM.

We understand that manufacturers are planning CD-ROM drivers that should make CD-ROM useable on our target range of work stations without requiring the use of `cd_link`. That will allow a user to do many things before a shadow tree is necessary.

A disadvantage of publication on UNIX machines is that superuser status is now needed to use the mount command. Thus an amateur cannot pop a disk in and out as simply as on a Macintosh. It is rumored that experts are working on this problem, but we can expect it to take some time before a solution is worked out for the many vendors we would like our software to work on.

I recently received a free CD called "Software Store" which contains demonstrations of a wide range of software for sale. I looked forward to putting trying it on my desktop workstation. A little booklet comes with the disk. Unfortunately the first four pages of this booklet are devoted to "super user" commands required to be run before we can see anything. I was first annoyed by learning that I needed to modify the "`/usr`" partition of my computer. To do so safely presumes the makers of "Software Store" have done everything correctly and that I can follow their instructions correctly. When I read further that I would need to modify "`/usr`" not only on my workstation but also on the file-serving computer that serves our entire group, then I postponed everything.

We recently purchased CD-ROM drives for workstations for about four hundred dollars each, so although the drives are not yet in widespread use, we expect them to be so soon.

The chore of organizing in general

The biggest headache of delivering and receiving software is the chore of installing it. Our goal is to make this easy for both sides. The Apple Macintosh has gone a huge distance beyond ordinary researchers in this respect, but I contend that we can learn how to deliver and receive software as easily as we can deliver and receive any electronic document. We have already learned how to maintain our programs on the five binary platforms at SEP (by using network file system and compiling with `cake`) and that covers most of the job. With the help of Wes Bauske and Mark Chackerian we are rapidly learning the remaining job—how to extract the relevant portions of our home system and make instructions on how it can all be built elsewhere.

OUR FIRST CD-ROM

We made three disks, known in the trade as "one-offs" which we denote "/sepcd1.alpha".

The contents of our first disk: sepcd-1.0.alpha

All the documents and program sources we can find are much smaller than the capacity of the disk. For example, my book text is about 1 Mb and its programs are about 3 Mb. All the software we share internally at SEP is perhaps 30 Mb. The needed data space is about 20 Mb. Typesetting software and fonts may be 50 Mb.

As you can see, there remains a huge space for other documents and data, more than we can use. To use some of the excess available space we decided to include program binaries for those computers that we have in our lab, IBM, Sun, and DEC. Thus most readers should not need to compile anything to start reading electronic documents and interacting with some of the figures. Hopefully we will need only a few written instructions to accompany the disk saying how to mount it and what to do then.

Experience with our first CD-ROM

Mark Chackerian, an undergraduate student here for the summer, did most of the day to day work preparing my book for the Young Minds Makedisk software. We were delighted that our first batch was useable. We kept one disk at Stanford and sent the other two around to friends and alumni at Chevron, IBM, Mobil, and Unocal who had varying degrees of success.

Theoretically, CD-ROM is slower than a hard disk, but by all reports, that does not seem to be detracting significantly from the appeal of the application.

As expected, installation is a significant problem. With perhaps one exception, the effort of installing was rewarded by satisfactory reading of the book with figures included. In many cases the figure pushbuttons didn't work. This partly represented difficulties people had during installation and partly represented some errors I had made with Mark. Two major headaches during installation have to do with X: does a complete X environment exist on the target machine? and can the fonts be found? This latter question particularly arises if the viewing machine is not the machine with the CD-ROM drive.

Outstanding results were obtained on IBM RS-6000, DEC-3100, and Sun workstations—except those Suns with GX accelerated graphics. Steve Cole modified our local software to avoid the postscript `setlinewidth` command; now we expect our CD to run on all Sun-4 models.

MARKETING

The CD-ROM is a publication, so we want to distribute it to our sponsors and also the public. Ideally, distribution to the public should pay for itself. But there are some special pricing problems with this publication. The manufacturing economics of CD-ROM is roughly this: the first one (the master) costs about a thousand dollars and after that they cost less than two dollars each. (It costs another dollar for the paper booklet). We can't predict how many will be distributed as a function of price. Historically, there have always been reasons for academics to subsidize their publications, and that seems especially the case here because the CD-ROM has little public acceptance yet. If we charge people $25 for the CD-ROM, then they have a right to expect it to work on their equipment, and we can't manage guarantees or consulting services. I have a magazine "Classical CD" with a CD attached to the cover. Perhaps we can plan on a mailing with such journals as Geophysics. This seems especially likely as we acquire more documents and the number of useable platforms grows. Meanwhile, Stanford University Press has agreed to sell CD-ROM copies of my book from our inventory at a modest price, while recognizing that I plan distribute as many free (or nearly free) copies as I can.

We hold no hostility to people who charge money for their software, indeed we like to purchase good software. But an unescapable consequence of publication via CD-ROM is that the author cannot use proprietary software that most readers might not have. When I applied this realization to my book, I found that I am using `ratfor` and `newawk`. Since then I adapted to public domain ratfor[3] and converted essential functions from `newawk` to old `awk` which is standard on UNIX systems.

Copyright, etc

The Free Software Foundation (also known as GNU), under the direction of Richard Stallman, has invented a novel mechanism for distributing software and other intellectual property. Their mechanism has proven to be popular with authors and much of the software that we use is available under GNU's conditions or similar ones. We get such software over the internet. This is a complex topic: Oversimplifying, the author claims copyright, but then the author goes further and offers you a license to copy the software if you abide by some stated conditions.

I am not sure if GNU's legal conditions are precisely what we want—especially the nickname "copyleft". But GNU has researched the legal side of this issue so thoroughly that I am choosing to use the "GNU general public license" which is attached.

I hope you will respect the legal rights of my paper book publisher and not make paper copies of the book. I hold the electronic rights. Please see the GNU General Public License which offers you various electronic rights. I would like you to be able to do what you like with my software, but my employer would like to benefit from any possible commercial applications you might come up with. If you want to make a product, I will help you communicate with my University's licensing office. They are nice people and I believe they accept reasonable offers. If your employer is a sponsor of the Stanford Exploration Project, you already have an option for a non-exclusive

[3]thanks to the help of Wes Bauske

royalty-free license. If your company is a sponsor of the Stanford Exploration Project, then you already have an option for a free license so then you can forget about the more restrictive GNU public license (as far as my software is concerned).

ACKNOWLEDGMENTS

I gratefully acknowledge the friendly advice of many people and the dedication and hard work of Martin Karrenbach, Mark Chackerian, Steve Cole, Dave Nichols, and Wes Bauske.

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 PERMIT-

TED 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 GNU 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) 19yy <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.

disC versus disK: A disc is a phonograph record or a CD-ROM with music on it and a disk is a CD-ROM with computer files on it.