Bil Reference Manual

The documentation for Bil 2.4
A modeling platform based on finite element/volume methods

Edition "1.6" (27 September 2016)

Patrick Dangla

Table of Contents

Copying conditions

Bil is "free software"; this means that everyone is free to use it and to redistribute it on a free basis. Bil is not in the public domain; it is copyrighted and there are restrictions on its distribution, but these restrictions are designed to permit everything that a good cooperating citizen would want to do. What is not allowed is to try to prevent others from further sharing any version of Bil that they might get from you.

Specifically, we want to make sure that you have the right to give away copies of Bil, that you receive source code or else can get it if you want it, that you can change Bil or use pieces of Bil in new free programs, and that you know you can do these things.

To make sure that everyone has such rights, we have to forbid you to deprive anyone else of these rights. For example, if you distribute copies of Bil, 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 tell them their rights.

Also, for our own protection, we must make certain that everyone finds out that there is no warranty for Bil. If Bil is modified by someone else and passed on, we want their recipients to know that what they have is not what we distributed, so that any problems introduced by others will not reflect on our reputation.

The precise conditions of the license for Bil are found in the General Public License that accompanies the source code (see Appendix A [License], page 23). Further information about this license is available from the GNU Project webpage `http://www.gnu.org/copyleft/gpl-faq.html`.

The source code of Bil (for Linux) can be downloaded from the webpage `http://perso.lcpc.fr/dangla.patrick/bil`.

If you use Bil, we would appreciate that you mention it in your work. References, as well as the latest news about Bil development, are always available on `http://perso.lcpc.fr/dangla.patrick/bil`.

1 Introduction

Bil is a modeling platform based on finite element/volume methods dedicated to coupled problems involved in environmental engineering, geomechanical engineering, material sciences, etc.. Bil is distributed under the terms of the GNU General Public License (GnuGPL). The source code can be downloaded at

`http://perso.lcpc.fr/dangla.patrick/bil`

Bil is intended to be used by students, engineers or reseachers to work out problems or develop their own models. Bil is written in C language. Thus provided that a C compiler was installed, it can run on any OS. However the complete installation of Bil (including documentations) can be achieved on Linux-based OS only (Debian, Ubuntu, etc..).

Bil is developped for 1D, 2D and 3D problems. It doesn't include a mesh generator and post-processing treatment of outputs. However it can read mesh files created by the open-source free software Gmsh downloadable at `http://www.geuz.org/gmsh/`. The output files created by Bil for 1D problems can be used easily by some plotting programs as Gnuplot. But as a general rule, post-processing options create output files which can be used by Gmsh for post-processing treatments (see Bil options).

2 Running Bil

Bil can be run non-interactively only. To compute the solution of a problem described in the input data file `my_file`, type:

```
bil [options] my_file
```

The input data file format is described in the following section. Without any option, running Bil will create the output files `my_file.pi` and `my_file.ti` (see Chapter 4 [Output files], page 9). If there is no input data file named `my_file`, Bil will open this file in writing mode and will ask the user for the input data. With the option -h (help) there is no input data file to give. The command

```
bil
```

(alone) displays the available options.

The way Bil will run may depend whether there are some other files (see Chapter 5 [Other files], page 11). The most important of them, namely `my_file.graph.iperm`, defines the way nodes are renumbered so as to optimize the resolution algorithm. Except in 1D problems, it is very important, if not essential, to create this file with an aquedate option, e.g. -iperm, (see Bil options).

3 Input data file format

The file `my_file`, as mentionned above, provides the input data of the problem to be worked out. A list of 4 capital letter key-words organizes the inputs in several groups such as mesh, material properties, boundary conditions, etc. For example the key-word `GEOM` is followed by the inputs defining the dimension and the symmetry of the problem. The full list of key-words which must appear in `my_file` is given in the table below. Any line begining with `#` is considered as comments and skipped.

key-word	**description**
`GEOM`	dimension and symmetry of the problem
`MESH`	mesh
`MATE`	material properties of the material index 1
...	...
`MATE`	material properties of the material index n
`FLDS`	fields i.e. space functions
`INIT`	initial conditions
`FUNC`	time functions
`COND`	boundary conditions
`LOAD`	loads
`POIN`	define some points for output files
`DATE`	define the dates for output files
`OBJE`	objective variations of the main unknowns
`ITER`	convergence criteria of iterative process
`TIME`	time steps calculation

An on-line help is provided by typing:

```
bil -h
```

4 Output files format

Each run produces 2 sets of output files. In the first set, output files are named:

`my_file.pi`

where *i* is an integer ranging from 1 to the number of points defined by the key-word `POIN`. There is no file if there is no points. These files provide the results obtained at the specified points. The first column contains the times at which the results have been obtained. The other columns contain the value of specific quantities as implemented in the model defined in the key-word `MATE`.

In the second set, output files are named:

`my_file.ti`

where *i* is an integer ranging from 0 to the number of dates defined by the key-word `DATE`. These files provide the results obtained at the specified dates. The three first columns contain the three coordinates of nodes. The other column contain the value of the same quantities as those contained in the first set of files.

Some lines of these 2 sets of files are commented as indicated by the character `#` in the first column. These comments provide some informations about the nature of the computed quantities found in the following lines.

5 Other files

Bil produces some files and sometimes can read and use some other files. The name of these files are formed with the name of the input data file and suffixes, `my_file.`*`suf`*, in the same way as the output files. They are listed in the table below.

fichier	**description**
`my_file.t`*`i`*	output files related to date index *i*
`my_file.p`*`i`*	output files related to point index *i*
`my_file.pos`*`i`*	view *i* to be read by Gmsh
`my_file.msh`	Gmsh mesh file
`my_file.graph`	mesh graph
`my_file.graph.iperm`	inverse permutations file
`my_file.sto`	storage file
`my_file.cont`	continuation file (see below for explanations)
`my_file.conti`	continuation file (see below for explanations)

The files `my_file.cont` and `my_file.conti` allow to continue a previous computation or to resume an interrupted calculation achieved with a previous input data file (`my_previous_file`). With `my_file.cont` the process doesn't go through the initialization stage (`ComputeInitialState`, see below) so that the calculation continues as if there hadn't been interruption. With `my_file.conti` the process goes through the intitialization stage so that some variables of the model can be re-initialized (e.g. strain variables can be reset to zero). To do so, copy the file `my_previous_file.sto` in `my_file.cont` (or `my_file.conti`) and run bil with `my_file` as a new input data file in which you will have defined some additional dates beyond the last date defined in `my_previous_file`.

6 Models

Mostly the concept of model refers to the constitutive equations or complementary laws that are needed to mathematically end up with a well-posed problem. However we need more informations, here, regarding the numerical methods that are used to handle complex problems and geometries. These methods will be implemented in a single file whose basename will identify the code name of the model. The concept of model will then be extended to fit with the set of informations pertaining to:

- The number and kind of equations to be solved
- The complementary laws that are needed to have a well-posed problem
- The informations associated to the numerical methods employed

Models aim at addressing the behavour of the material at the scale of one finite element. Therefore the methods defined in object "Model_t" aim at computing matrix, residual forces, outputs and so on, for the nodes of one element. The object "Element_t" is therefore the main input entry of all methods of object "Model_t".

A short description of the available models can be displayed by typing:

```
bil -m
```

7 How to develop a new model ?

To tell Bil to account for a new model, you just need to create a new file, e.g. `my_model.c`, in the folder `ModelFiles` and add the basename of this file, namely `my_model`, to the list of the available models found in `ListOfModels.inc`. This model will be taken into account automatically the next time binary files will be created.

To help you in creating this new file, it is recommended to learn from already existing files. This file should contain at least the 11 methods of the Model class-like structure. These methods are listed in the table below.

method	**description**
`SetModelProp`	Set the model properties
`ReadMatProp`	Read the material properties
`PrintModelProp`	Print the model properties
`DefineElementProp`	Define some properties of the element.
`ComputeInitialState`	Compute the initial state
`ComputeExplicitTerms`	Compute the explicit terms
`ComputeMatrix`	Compute the matrix
`ComputeResidu`	Compute the residu
`ComputeLoads`	Compute the loads
`ComputeImplicitTerms`	Compute the implicit terms
`ComputeOutputs`	Compute the outputs

8 Examples

8.1 Drainage of a column

This problem is governed by the Richards' equation. A 1 meter high sand column is initially satured. The liquid pressure is initialized as: $p_l = p_{atm} - g(x-1)$. At $t = 0$ we drained the column from the bottom by imposing the pressure to $p_l = p_{atm}$. The input data file is given below.

inputs	**comments**
`# Drainage of`	You can write some comments
`# a sand column`	by beginning any line with #.
`GEOM`	Geometry of the problem
`1 Plan`	1D pb, plane symmetry
`MESH`	Mesh
`col.msh`	The mesh is read in this file (format Gmsh). This mesh consists in a 20 elements mesh between 0 and 1. There are 2 regions. The region 1 is the point at 0. The region 2 is the line between 0 and 1. There is 1 material.
`MATE`	Material 1
`Model = m1`	code name of the model
`gravite = -9.81`	gravity
`phi = 0.3`	porosity
`rho_l = 1000`	fluid mass density
`k_int = 4.4e-13`	intrinsic permeability
`mu_l = 0.001`	fluid viscosity
`p_g = 100000`	gas pressure
`Curves = tab`	in the file tab, there 3 columns: p_c S_l k_{rl}
`FLDS`	Fields
`2`	2 fields
`Type = affine Value = 1.e5 Gradient = -9.81 Point = 1.`	affine field defined by $10^5 - 9.81 * (x-1)$
`Type = affine Value = 1.e5 Gradient = 0. Point = 0.`	constant field equal to 10^5
`INIT`	Initial conditions
`1`	1 initial condition
`Region = 2 Unknown = p_l Field = 1`	in the region 2, $p_l = 10^5 - 9.81 * (x - 1.)$.
`FUNC`	Time functions $f(t)$
`0`	here there is no function
`COND`	Boundary conditions
`1`	1 boundary condition
`Region = 1 Unknown = p_l Field = 2 Function = 0`	in the region 1, $p_l = f(t)*10^5$ (by default $f(t) = 1$)

`LOAD`	Loads
`0`	there is no load
`POIN`	Points where we want outputs
`0`	no points
`DATE`	Dates where we want outputs
`2`	2 dates
`0. 1800000`	$t_0 = 0$ and $t_1 = 1800000$
`OBJE`	Objective variations of unknowns
`p_l = 1000`	objective variation $\Delta p_l = 1000$
`ITER`	Parameters for the iterative process
`Iterations = 20`	20 iterations
`Tolerance = 1e-10`	the tolerance is 10^{-10}
`Repetitions = 0`	no repetition
`TIME`	Parameters for time steps calculation
`Dtini = 1`	initial time step equal to 1.
`Dtmax = 3600`	maximum time step equal to 3600.

9 Versions

News in 2.3: Use of C++ compiler. Programmation is now extended to C++ langage. Hence the use of .cpp and .hpp files is allowed.
A new object "Exception.h" has been created to handle exception mechanisms such as interruption, floating point error.
In such occuring event the program handles the event, saves outputs and exits in a clean way.
Introduction of the curve builder "Expressions", based on the evaluation of mathematical expressions obtained from AnaGram (www.parsifalsoft.com).
In this version a shared library, libbil.so, from the genuine sources of Bil is created and installed in the machine. Linkage with other possible external libraries is possible. A new extension of input file ".conti" has been introduced so that we can load the solution from a previous calculation and continue this calculation while going through the initialization stage (see documention).

News in 2.2: WARNING, the outputs of FVM_ComputeIsotropicConductionMatrix() and FVM_ComputeMassAndIsotropicConductionMatrix() have been modified see examples in models using these methods. New objects have been introduced: Views, TextFile, CurvesFile, MatrixStorageFormat. Introduction of the curve builder "Evaluate", based on the evaluation of mathematical expressions obtained from Snippets (www.brokersys.com/snippets/).
Some bugs fixed, e.g. those in "Buffer_FreeFrom". Extended field delimiters in curves files. Some new models.

News in 2.1: A class-like structure Buffer_t has been introduced as a circular buffer. It can be used in any functions to compute vectors or tables of any type. Numerical Methods, such as FEM or FVM, have been implemented in files such as FEM.[c,h] and FVM.[c,h]. These methods are viewed as object with class-like structures FEM_t and FVM_t.
In these files methods have been implemented such as FEM_ComputeMassMatrix and can be used directly in models provided that header has been included. The folders "Common" and "Main" have been created.
We moved the main files in "Main" and files of common use, like "Buffer.c", in "Common". DataBases have been created which can be used in model files.
The matrix storage format have been implemented in a separate file "MatrixStorageFormat.h".

News in 2.0: The code has been rebuilt to an object-oriented programming code. However the code is still implemented in C language. Class-like structures have been introduced and implemented in separated files.
Each class-like structure has attributes and pointers to functions (see e.g. structure Model_t). Moreover some improvements have been introduced. The file names of models (in the folder "ModelFiles") can be chosen abitrarily. The class-like structures "NodeSol_t" and "ElementSol_t" are two linked lists which contained the nodal and elemental solutions. A new pointer to "double" type in "ElementSol_t" can be used to store constant terms.

Nouveautes dans 1.8: creation d'une structure "modl_t" contenant les methodes (taches elementaires) i.e. des pointeurs sur fonctions.

Nouveautes dans 1.7: option "base" ajoute dans le Makefile. Qqs nouveaux modeles. Option "Relative" possible dans OBJE, donnee apres la valeur. Chargement automatique des modeles, il n'est plus besoin de modifier le fichier "xmod.c". Les fonctions "dm,qm,tb,ch,in,ex,ct,mx,rs,so" sont renommees "dmNB,qmNB,tbNB,chNB,inNB,exNB,ctNB,mxNB,rsNB,soNB". Les titres des modeles ainsi que des exemples de donnees sont geres par les fonctions "qm1, qm2 ...". Options nouvelles de la ligne de commande. "bil -m" affiche les titres des modeles. "bil -m I" affiche un exemple de donnees du modele I. Possiblite d'avoir des elements d'ordres multiples (1 pour chaque inconnue). Cette possibilite est geree par les tableaux "el.pin" et "el.peq". Une valeur negative associee a un noeud et une inconnue n'est pas prise en compte comme inconnue globale. Possibilite de creer de nouvelles fonctions d'interpolation au niveau des modeles dans "tb1, tb2, ..." avec la fonction "creer_interpolation(...)". Possibilite de definir des champs aux points d'intersection d'une grille dans l'espace avec l'option "Type = grille" dans CHMP (l'option Type = affine restant par defaut).

Nouveautes dans 1.6: bug corrige dans "sauvep". Les fichiers "mod.c" et "mod.h" sont renommes "xmod.c" et "xmod.h". Le specificateur de type "void" est remplace par "int" pour les fonctions "ex1(), ex2() ..." definies dans "m1.c, m2.c ..." et pour l'identificateur de type "ex_t" defini dans "defs.h". On utilise ce retour dans les fonctions "explicite()" et "algorithme()" definies dans "calc.c". Compatibilite avec le format de maillage version 2.0 de GMSH.

Nouveautes dans 1.5: definition des types de fonctions "dm_t, "qm_t", "tb_t", etc... Reorganisation du fichier "mod.c" par l'introduction de tableaux de pointeurs de fonctions pour simplifier l'introduction de nouveaux modeles. Possibilite de charger la librairie de SuperLU (voir le fichier "make.inc"). Creation de l'option de la ligne de commande "-m slu" qui permet d'utiliser la methode de resolution proposee par SuperLU a condition d'avoir construit, auparavant, le fichier des permutations inverses a l'aide d'un programme adapte comme Metis.

Nouveautes dans 1.4: reorganisation et creations des fichiers : calc.c, lecdo.c, postt.c, renum.c. Suppression des variables statiques. Creation d'une structure de donnees "dnns_t", d'une structure pour la matrice "mtrx_t", d'une structure pour la solution "sltn_t", d'une structure pour la renumerotation "nume_t". Amelioration de certaines fonctions de lib.c. Suppresion de rssurf (pris en charge par rsmass).

Nouveautes dans 1.3: nouveaux modeles inclus. Qqs ameliorations apportees dans le calcul de dt (fonction pasdt). Apport de nouvelles info dans les structures mate_t, elem_t: en particulier neq,eqn,inc dans mate_t. Creation de la structure node_t et suppression du pointeur no_x. Cette version permet de prendre en compte des modeles reposant sur un nombre d'equations et sur des natures d'inconnues pouvant varier d'une region a l'autre. Cette version realise la continuite des equations en fonction de nouvelles informations contenues dans les modeles comme le nombre d'equations et les noms (predefinis) de chaque equation et inconnue associee. En consequence la structure du fichier a ete (legerement) modifiee pour prendre en compte ces informations, notamment dans les mots-cles INIT,COND,CHAR,ALGO (voir l'aide en ligne). Pour plus de clarte les donnees relatives aux variations objectives des parametres ont ete deplacees du mot-cle ALGO dans le nouveau mot-cle OBJE.

Nouveautes dans 1.2: creation d'un repertoire exemples. Simplication de la gestion des elements de surface (suppression de SURF). Creation de champs (mot-cles CHMP). Gestion des interruptions dans le calcul de la matrice.

Nouveautes dans 1.1: creation des pages info et de la doc sous differents formats (ps,pdf,txt). Creation des repertoires bin et lib. Amelioration de la procedure d'installation. Simplification du jeu de donnees par la definition des regions de maillage.

Appendix A 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.

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

1. 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.
2. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

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

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

```
one line to give the program's name and a brief idea of what it does.
Copyright (C) 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.