Bil Reference Manual

The documentation for Bil 2.11 (27 October 2025)
A modeling platform based on finite element/volume methods

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://bil.ifsttar.fr`.

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://bil.ifsttar.fr`.

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://bil.ifsttar.fr`

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 like Windows, MacOS or Linux-based OS.

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 such 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). 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 storage of the LU decomposition of the matrix, in case of a Gauss elimination method (Crout's method). Except for 1D problems where nodes are numbered from one end to the other of the mesh, it is very important, if not essential, to create this file with an aquedate option, e.g. -iperm, (see Bil options). In case of multi-frontal methods (such as SuperLU or MA38) or Krylov space methods (such as those found in Petsc library), this file is not needed.

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 reserved key-words (with an upper-case first letter) organizes the inputs in several groups such as mesh, material properties, boundary conditions, etc. For example the key-word `Geometry` is followed by the inputs defining the dimension and the symmetry of the problem. Any content of a line after `#` is considered as comments and skipped. The full list of reserved key-words which must appear in `my_file` is given in the table below.

key-word	**description**
`Geometry`	dimension and symmetry of the problem
`Mesh`	define the mesh
`Material`	material properties of the material index 1
`...`	. . .
`Material`	material properties of the material index n
`Fields`	define fields i.e. space functions
`Initialization`	define the initial conditions
`Functions`	define time functions
`Boundary Conditions`	define the boundary conditions
`Loads`	define the loads
`Points`	define some points for output files
`Dates`	define the dates for output files
`Objective Variations`	objective variations of the main unknowns
`Iterative Process`	convergence criteria of iterative process
`Time Steps`	time steps calculation

Some other optional input data can also be provided:

key-word	**description**
`Units`	change the units
`Periodicities`	periodicities of the mesh for peridiodic problems
`Model`	the names of unknowns can be changed

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.p`*i*

where *i* is an integer ranging from 1 to the number of points defined by the key-word `Points`. If there is no point no file will be created. 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 `Material`. In the second set, output files are named:

`my_file.t`*i*

where *i* is an integer ranging from 0 to the number of dates defined by the key-word `Dates`. 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.

file	**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`. Boundary conditions can also be modified in this new input data file in order to deal with changing from Dirichlet to Neumann boundary conditions.

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 are implemented in a single file whose basename identifies the code name of the model. Therefore the concept of model should 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 behaviour 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 `src/Models/ModelFiles` of the Bil's source directory and add the basename of this file, namely `my_model`, to the list of the available models found in `src/Models/ListOfModels.inc`[1]. 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

[1] Upper camel case convention is used to name the source files of this project. The same convention is used for the attributes and the methods of objects.

8 Examples

8.1 Drainage of a column

This problem is governed by the Richards' equation. A 1 meter high sand column is initially saturated. 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 #.
Geometry	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.
Material	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}
Fields	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
Initialization	Initial conditions
1	1 initial condition
Region = 2 Unknown = p_l Field = 1	in the region 2, $p_l = 10^5 - 9.81 * (x - 1.)$.
Functions	Time functions $f(t)$
0	here there is no function
Boundary Conditions	Boundary conditions
1	1 boundary condition
Region = 1 Unknown = p_l Field = 2	in the region 1, $p_l = f(t)*10^5$ (by default $f(t) = 1$)
Function = 0	

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

9 Version history

News in 2.10: Reorganization of the header files regarding the dependencies that only require to be forward declared and those that need to be #included.

News in 2.9: Use of the library "autodiff" via option at compiling time.
Compliler c++ is forced for all source files (.c and .cpp).
Introduction of the interface "MaterialPointMethod.h" for an easiest way of implementating models.

News in 2.8: Parallelization via OpemMP and MPI.
A new algorithm is implemented in the module SNIA which is based on a sequential non iterative approach (SNIA). The build system is switched to CMake. Bugs fixed in MA38 solver. New way of using extra libraries.
The number of temporal discrete solutions kept in memory for possible use in time discretization schemes can be given in the command line.
The solver KSP from the Petsc library can be selected if the external library "libpetsc.so" was installed (see EXTRALIBS).
Re-arranging some allocation of memory space for the mesh.
Reformulation of the jacobian matrix of the transformation between the element in the reference frame and that in the current frame. Zero-thickness element can be dealt with. Unknowns like pressure/temperature or other can be made continuous across such zero-thickness elements. The python script "utils/gmshcrack/gmshCrack.py" can be used to insert zero-thickness elements in mesh file generated by GMSH.

News in 2.7: Improvements on exception mechanisms, in no case results will be lost.

News in 2.6: The solver of sparse unsymmetric system of linear equations MA38 from HSL is introduced in the package and can be selected from the command line.
The solver SuperLU can also be selected if the external library "libsuperlu.so" was installed (see the file "make.extralibs").

News in 2.5: A new interface "Session.h" is created. Methods like Session_Open and Session_Close can be used to open and close sessions.
Sessions are in a stack with the current session at the top.
Open a session = push it, close a session = pop and delete it.
If needed instances like Message, Exception, etc... are created once in each session of the stack. Only those of the top session are currently used.

News in 2.4: Two new database interfaces: "HardenedCementChemistry.h" and "CementSolutionDiffusion.h" to be used for cement-based materials.
New models now use these interfaces. The references used to implement some databases are now copied in the folder "src/Models/DataBases/References".

News in 2.3: Use of C++ compiler. Programmation is now extended to C++ language. 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 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. 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 them 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 prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. 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.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

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

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey 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; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If 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 convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU 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 that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

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.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS 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.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

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 state 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) {year}  {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 3 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, see <http://www.gnu.org/licenses/>.
```

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

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

```
    {project}  Copyright (C) {year}  {fullname}
    This program 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, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU 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 Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.