GNU Source Installer

version 0.5, 16 September 2005

Claudio Fontana claudio@gnu.org

Published by the Free Software Foundation,
51 Franklin Street - Fifth Floor,
Boston, MA 02110-1301, USA

1 Introduction for Beginners

This is an introduction for real beginners of source installation. If you are already somewhat experienced with UNIX-like systems and GNU source code install procedures, you should skip this chapter, otherwise

Welcome!

`sourceinstall` will try to make configuration, compilation, installation and removal of source packages easier for you as a beginner.

Instead of hiding information and operations from you, everything that happens will be available for you to see.

This way, if you are interested you can hopefully understand basic concepts by just looking at the commands executed by this program in the information frame.

Provided your system meets the requirements, and thus you manage to have a working installation of `sourceinstall` itself, you will be shortly able to install new software from its source code by surfing the web, identifying a Free Software you like, downloading its SOURCE package, and feeding it to GNU Source Installer.

1.1 Installing the Installer

If you have a fast internet connection, proceed to download sourceinstall-0.5-fullpack.sh or a newer version, and mark where the file will be placed (the "folder", the *directory*). You must have the permission to write in that directory. For example, we will assume that you are downloading to

```
/home/user/downloads
```

After the download completes, start a console session. Your desktop environment should include a button, picture, or menu item that refers to a "shell", "terminal" or "console".

After the console is open, you should see a brief message ending in $; this message will be represented here by a single dollar character, and you shall NOT type that character as part of the commands.

Reach the directory you just downloaded your file in, by typing in the shell this command, followed by a RETURN:

```
$ cd /home/user/downloads
```

where of course '`/home/user/downloads`' is the directory in which you downloaded the file.

If you get an error message, double check your command for typos. If things are going well, you will know because you will get nothing else than another $ ended message (a *prompt*).

At this point, you can decide if you want to install as the super-user (root), or using your ordinary account.

Installing as root is more indicated for system-wide installs. To do so, type

```
$ su
(Enter your root password)
```

At this point run the installer by typing:

$ /bin/sh sourceinstall-0.5-fullpack.sh

wait for the package to extract (this might take some minutes on slow or loaded systems), and follow the instructions. After being asked some simple questions, hopefully you will get a working installation of the GNU Source Installer. Mark the executable name that is showed at the end of the procedure, because that is the program that you need to run to start the installer.

This setup procedure installs in a subdirectory of your home directory by default if you are using an ordinary account. In particular, by default the installation *prefix* (the directory subtree in which to install) is ~/usr, where ~ represents your home directory.

If you are using a root account, then the setup procedure will instead use /usr/local as the default prefix. This is a common prefix for system-wide installs.

If you have a fast internet connection and want the easiest install, you can skip the rest of this section, and jump to the "Troubleshooting" section if you experiece any problems during installation.

If you are a bit more daring, instead, you can try a normal source release. The releases whose names end in -fullpack are many megabytes in size, because they contain all the major dependencies in source form, and it could be that you already have the required packages.
You can fetch the much smaller sourceinstall-0.5.tar.gz (or other version). We will assume the same destination directory as above for the download.

At this point enter the following command (this assumes you have the GNU version of `tar`):

```
$ tar -zxvf sourceinstall-0.5.tar.gz
```

or, if you do not have the GNU version of `tar` (f.e. Solaris):

```
$ gunzip sourceinstall-0.5.tar.gz
$ tar -xvf sourceinstall-0.5.tar
```

Note that you can use the tabulation (TAB) character to complete names. Experiment with pressing (TAB) around the middle of the file name.

All the files you'll see are being extracted from the archive, and a new directory is being created in the current one. At the end type:

```
$ cd sourceinstall
$ ./configure
```

A lot of output will be showing at this point. The software is being *configured* (adapted) for your system. If everything runs smooth, you will see at most WARNINGs but no ERRORs. After a while you will get the familiar dollar, and now you can write:

```
$ make
```

Some output will be shown, then again the familiar prompt. And now:

```
$ su
(Enter your root password)
# make install
```

After writing `su`, you will be asked for your root password. You should have set your root password during your Operating System initial setup. If you do not know, try pressing ENTER. The # character before `make install` denotes the fact that after `su` you have gained root privileges. Since you are done, drop your root privileges by typing

```
# exit
```

Each time you want to run GNU Source Installer from the console, type:

```
sourceinstall
```

To run it from the graphical environment, you should create some kind of "shortcut" or "link" to the program on your desktop or in your program menus. The program to launch is (assuming a root installation and default values) '`/usr/local/bin/sourceinstall`' .

If you experience errors that prevent the correct installation and execution of the program, the next section tries to deal with these cases.

At the end of the procedure you will get the following files installed:

1. '`/usr/local/bin/sourceinstall`' (link to the program)
2. '`/usr/local/bin/sourceinstall.tcl`' (the program)
3. '`/usr/local/info/sourceinstall.info`' (texinfo manual)
4. '`/usr/local/man/man1/sourceinstall.man`' (man page)

To consult the GNU Source Installer manual type

```
$ info sourceinstall
```

If you want a brief overview of program invocation and options, you can consult the traditional man page by issuing

```
$ man sourceinstall
```

1.2 Troubleshooting Installation

If you could not install GNU Source Installer, this is most likely because you do not have the required software in your system. In other cases, it could be a bug in the installation procedure.

For the -fullpack releases, it is most likely the second (the special -fullpack release is meant to install without dependency errors).

If you need to report a bug that prevented the correct installation of a -fullpack, please provide all the files ending in .log and .err that are generated during the procedure.

To find all these .log and .err files, after getting the error during installation, look in the directory in which you ran the procedure. In our example, it was:

'`/home/user/downloads`'

You should find a '`sourceinstall-fullpack`' directory there. It is a directory created during installation, that is removed after the operation and only if it completed correctly. In your case, if you have an error during the procedure, the directory will still be there. That directory should contain many files whose names end in .err and .log . These are the files to attach to your bug report. As always, report your bugs to `bug-sourceinstall@gnu.org` .

If you are using a normal release (not a fullpack), look at the output of `./configure`, instead, and you will see if some needed programs have not been found on your system.

The most blocking thing is if you miss `tcl`, `Tk` or `Expect`: in this case `./configure` will exit with an error, and `make` will not be able to run.

If this is your case, you will need to install tcl, tk and/or expect (or try the -fullpack release instead). If you have your OS installation disk(s), chances are that the software is available there, and is installable using the OS specific installation system.

Otherwise you should fetch and build the Tcl, Tk and Expect source packages. You can fetch Tcl and Tk from `http://sourceforge.net/projects/tcl/`, while Expect is available at `http://expect.nist.gov/` . They are a bit tricky to install, so if you cannot find your way out of them, you can always revert to the -fullpack release.

If you succeed in building a working Tcl/Tk/Expect environment, restart the procedure from `./configure` and things should be better.

If you miss any of the other helper programs, only the particular functionality offered by that program will be missing (a WARNING will be shown).

This is a comprensive list of programs that GNU Source Installer uses, from the most important ones, to the really secondary:

- tcl interpreter `tclsh` with the Tk and Expect tcl packages, or in alternative the all-in-one `expectk` interpreter. If Tk is not present, you will still be able to use sourceinstall through the command line, but trying to launch the graphical interface will report an error.
- sh, cp, mv, rm, du, su, rmdir, ln, find, make These should be provided by your Unix-like OS. You must have them for GNU Source Installer to work correctly.
- tar this is necessary to extract the source .tar archive - in theory you could use GNU Source Installer without this, but in practice you will need it.
- gzip, gunzip these are necessary to uncompress .gz files and to compress the stored source using the quick and space-efficient .gz compression.
- bunzip2, bzip2 these are necessary to uncompress .bz2 files and to compress the stored source using the very space-efficient (but more time consuming) .bz2 compression. By default, the source is stored using this compression.
- ps this command is used to better detect other sourceinstall processes. GNU Source Installer also uses a lock file, so it is not really necessary.
- unzip, zip these deal with .zip files. It is very unlikely that you will find Free Software source packages available only in .zip format so you can ignore this. Also, you do not want to compress your source using zip. I found both .gz and .bz2 to be more efficient at least when dealing with source. Ignore warning if not present.
- compress this deals with .Z files. The gzip utilities are used to uncompress .Z packages, so you would only need this if for some arcane reason you want to store source in .Z format. Ignore warning if not present.
- chmod, chown these are used by the very experimental (and pointless?) binary package export feature.

If you still have problems, you can write an email to the `sourceinstall` users mailing list `bug-sourceinstall@gnu.org` and ask for help. Something that can help a lot is reporting the full output of

```
$ ./configure
```

Feedback of any sort is also welcome. Good luck, and I hope this helps :)

2 Introduction for the Experienced

GNU Source Installer (`sourceinstall`) is a complete source package management tool, that handles source package configuration, installation, tracking and removal, information querying and export.

It is intended to work on modern Unix-like systems (with `GNU/Linux` as a primary target).

This is a tool intended for the user, not for developers: it has nothing to do with package creation. It helps people install and manage software packages in source form.

The user installs new source packages by browsing the web, downloading a source package (in .tar.gz or other formats), and then feeding it to the source installer.

There are two different ways to interact with `sourceinstall`: a Tk graphical interface, and a command-line driven, non-interactive interface.

2.1 Why should you use this program?

If you already build most of the software on your system from source code, you might try `sourceinstall` nonetheless. Here is what this software offers to the experienced user:

- a centralized way of handling source installations

  even if you have no problems building from source code, you can use `sourceinstall` as a way to centralize and better organize your source installations.
- a GUI and a complete command line interface,

  that should satisfy the needs of many. The GUI interface also shows all output of the underlying command line tools called, and displays information about installed packages and available actions and preferences, so nothing will be hidden from you. The command line interface and the GUI are functionally equivalent.

  Scripts can interface with GNU Source Installer at the executable level, by using the command line interface.
- tracking of all currently installed source packages,

  with info about install size, source size, and all files relevant to each package.
  Yes, you can build everything from source directly using ./configure, make, make install, but `sourceinstall` helps you remember which configure parameters you used back then to configure that package, helps you after installation by tracking the executable to run, the available documentation, etc from a handy file list, offers editable package descriptions, and other features.
- automatic quick check for broken installs.

  If a file associated with an installed package goes missing, then (Tk interface) clicking on the package will prompt the problem, and mark the missing file with big warnings. If you can not solve the issue yourself by restoring the file, you can ask for a fresh reinstallation by clicking “Reinstall”. The command line ‘`--check`’ and ‘`--install`’ actions can be used for the same goal.
- implicit check for reliable and gnu-conforming packages.

  The program will issue warnings if the package offers only a very spartane build system, or does not correctly honor common Makefile targets and features.

- full debugging output.

  You will be able to get notified of warnings and errors from `./configure`, `make`, and all other programs.
- clean uninstallation.

  The program performs crosschecks between make uninstall results and internal information available on individual packages (gathered during the install process in a clean and portable way). These checks can detect files left over by the make uninstall procedure, and if no other package claims them, they are suggested for removal (Tk interface), or removed directly (cmdline interface).
- portability.

  The program should work on most modern Unices. Even though great care has been taken as to use only very portable code in both the program and its build system, something can slip by, even more since I (the author) do not have other machines than a trusted GNU/Linux box. Just report them: chances are, you will get the portability problem solved sooner than expected.

2.2 Why should you avoid this program?

The following is a critique against this tool, that shows what you lose, or do not gain in contrast to relying for example on the good old command line:

- you can lose time.

  `sourceinstall` basically installs two times. The program makes a test installation first (if `DESTDIR` or `INSTALL_ROOT` are supported), to make a final check and gather useful information. This has an impact on total installation time. It is a necessary overhead to avoid non-portable low level solutions. In addition, other phases you can sometimes skip when running the commands yourself will add up to the total installation time.
- some packages might not work.

  `sourceinstall` has to make some generalizations and will not be able to install difficult packages. An experienced user or developer can quickly go through a broken or sketched `Makefile` and fix things for his system, but `sourceinstall` can not. Also, packages which use different installation conventions (for example imake), do not work with `sourceinstall`. This program actively supports the autotools and the derived build system. If enough people use this tool, this could further drive developers towards the autotools and to create better packages in general.
- no package-level dependency tracking, no repository, nothing at all.

  This is not a GNU/Linux distribution. It is a source package management tool. If your package blocks during configuration, you still look at that error message in the console or pseudo-console and act consequently (generally this involves browsing for that missing file/package). This program does not interact with a repository of “installable” packages and dependencies.
- people who install everything from source might not give a damn.

  If the benefits shown in the preceeding list do not apply to you, you might prefer to just install from source as you always did. After widespread adoption, however, even less experienced users than you could be able to approach the source packages (and hopefully become more experienced with time).

3 Invoking

This chapter shows the various actions, options, parameters and preferences that `sourceinstall` accepts.

3.1 Synopsis

```
sourceinstall

sourceinstall  ACTION [ ACTION_ARG... ]
               [ OPTION [ OPTION_ARG... ]... ] [ PACKAGE_NAME ]

sourceinstall  [ OPTION [ OPTION_ARG... ]... ] FILENAME
```

`sourceinstall` can be run in three different ways.

If called without any arguments, the program starts a graphical interface and waits for user input to decide the action to perform. This is what beginners should probably do.

The second way of calling the program is the complete command line interface, where a single action must be specified, followed by zero or more options, possibly followed by a final '`PACKAGE_NAME`' if the action requires it.

The third way of calling the program is without specifying any actions, and with a required '`FILENAME`'.

This is for both convenience and backward compatibility, and is a shortcut for the '`--add`' action with no custom package name.

3.2 Actions

Only ONE action can be specified on the command line. The action may require one or more parameters.

Some actions require a '`PACKAGE_NAME`' on which to act, while others behave differently if such a '`PACKAGE_NAME`' is specified.

For example, the '`--export`' action can export information about all installed packages or about a single package, depending on whether a '`PACKAGE_NAME`' is specified.

As another example, the '`--add`' action adds a new package to the `sourceinstall` package list (a very common operation). If '`PACKAGE_NAME`' is specified, then the new package will be called '`PACKAGE_NAME`'. Otherwise, a default value is obtained from the top source directory of the package.

'`-h, --help`'
: show brief command line help

'`-V, --version`'
: show program version

'`-l, --list`'
: list existing packages;
if PACKAGE_NAME is specified, it is treated as a regular expression, and only

matching package names are included in the output. The regular expression is a tcl ARE. Tcl regular expressions have been implemented using the package written by Henry Spencer, based on the 1003.2 spec and some (not quite all) of the Perl5 extension.
Packages are separated by newlines, and a short description is shown near the package name if such a description is available. Package name and description are separated by a " - " (space, dash, space) sequence.

You might be tempted to pipe the output to a pager. If you do, and have expect-5.43 installed, avoid `more`. There is a bug in that Expect version that causes problems with pipes, and you will get a *broken pipe* error with more. Expect-5.41 does not show this problem.

'`-a, --add FILENAME`'
: add new source package from FILENAME;
this is the way to add and install new source packages, so you will use this often. Note that this is not the same as the '`--install`' action, which only acts on an already added (but uninstalled) package.

The '`--add`' action configures, builds and installs the source code referenced by FILENAME, then stores the configured source code and saves package information. The unique package name will be taken from the top source directory by default, but can be specified using PACKAGE_NAME (look at the SYNOPSIS above).

Note that since the source code is automatically stored (in the '`~/.sourceinstall/src`' directory) you can safely remove FILENAME after a successful '`--add`' action.

All the above assumes the default behaviour. It can be altered by Options and Preferences.

'`-r, --remove`'
: remove source package PACKAGE_NAME;
by default, the '`--remove`' action removes both installation and stored source code. In this case, after removal the package will not exist anymore, and its name will not show in the package list. You can instead fine grain what gets removed by using the '`--binary`' or '`--source`' options.

'`-c, --check`'
: show package PACKAGE_NAME information;
if the package is currently installed, validation on known installed files is performed, signaling eventually missing files. If the check reports some missing files, you can rely on the '`--install`' action to reinstall over the broken installation.

'`-v, --update OLD_PACKAGE FILENAME`'
: remove package OLD_PACKAGE completely, then add FILENAME as a new package.
As for the '`--add`' action, the new package name will be taken from the top source directory by default, or from PACKAGE_NAME if specified.
This action is nothing more than a twofold '`--remove`' / '`--add`' action.

'-i, --install'
: install known package PACKAGE_NAME from stored source code.
This action is not the way to deal with a new package. If you want to add a new package, you want to use the '--add' action instead.

'-u, --uninstall'
: this is an alias for -r -b .
This action only removes the installation, but not the source code. Useful to remove something you might wish to restore at a later date. If you want to remove and completely forget about a package, use the '--remove' action instead.

'-d, --description STRING'
: associate first line of STRING to package PACKAGE_NAME short description, and the remaining lines to the long description.
If STRING is empty, remove descriptions from the package.
To submit an empty string, or a string containing newlines, make use of your shell quotation characters. Two examples (bash):

```
$ sourceinstall -d "STFU - Shoot The Fighters Up space game
> This game is the famous SDL based STFU space game, where lots of noisy
> enemy fighters dance through the screen making every kind of disturbing
> crappy sound. Using overpowered weapons, you can finally bring them to
> silence, and restore peace to the galaxy." STFU
```

Here we add a package description to the package STFU. The first line will be the short description, and the rest of the multiline text is called the long description. The descriptions will show everytime the '--list' and '--check' actions are requested (see those actions for further info).

```
$ sourceinstall -d "" STFU
```

Here we remove all the descriptions (both short and long) from the STFU package. The '--list' and '--check' actions will not show any descriptive text anymore.

'-n, --rename OLD_NAME'
: rename existing package OLD_NAME to PACKAGE_NAME.

'-x, --export FORMAT FILENAME'
: export package information to FILENAME.
FORMAT can be xml, txt or lst . The xml format and txt format contain all the package information, while the lst format only exports the installed files list (and thus has only sense when applied to an installed package).
If a PACKAGE_NAME is specified, then only the information regarding PACKAGE_NAME is exported. If no PACKAGE_NAME is present on the command line, information about all known packages is exported (xml and txt formats only).

'-p, --pack FILENAME'
: build binary tarball from the installed package PACKAGE_NAME, and save it as FILENAME.
Permissions and ownership are preserved. This functionality is experimental,

unstable, slow, sketched. You will need to have or acquire root privileges to complete this operation.

3.3 Options

Many options can be specified on the command line, but each option can be chosen only once. Options modify specific actions' behaviour.
Some of these options overlap with the preferences in the sourceinstall configuration file, and could even overlap with the package information. These command line options take precedence in these cases.

'`-s, --source`'
: apply add or remove action to source only.
During an '`--add`' action, use this option to specify that you do not want the software to be installed, but only want to store the source internally for future needs. You can combine this with the '`--configure`' option to prepare a configured source package and then store it for later use.
During a '`--remove`' action, use this option to specify that you want only the source code to be removed. If the package is currently installed, this leaves only the binary installation, without the stored source code that generated it. If you have enough disk space, it is recommended to keep the source code.

'`-b, --binary`'
: apply add or remove action to installation only.
During an '`--add`' action, use this option to skip the "store source code" phase. If you have enough disk space, it is recommended to keep the source code.

'`-t, --strip`'
: strip binaries during installation if possible (not recommended)

'`-z, --compression FORMAT`'
: use FORMAT as compression format for storing source code. Can assume values gz, bz2, Z .
This overrides the '`SRC_COMPRESS`' preference .

'`-C, --configure STRING`'
: (re)configure package using STRING.
First char of STRING cannot be a '-'. This is worth saying because a common usage of the '`-C`' option is to override the default installation prefix set in the Preferences. The command line parser will take your "–prefix" string for another option and will complain about a missing '`-C`' parameter. To avoid this, just prepend your configuration string with a harmless space.
You can take the habit to prepend a space to every configuration string you submit, and you will be fine. Here is an example:

```
$ sourceinstall --install -C " --prefix=/home/claudio/usr" unshield-0.5
```

Note that if you want to submit some environment variables to configure, a nice way to do it is to specify them in the configure STRING. This way they will appear in the package information. For example:

```
$ sourceinstall --add cmdftp-0.9.2.tar.gz -C " CFLAGS=-Os LDFLAGS=-s"
```

'-D, --subdir STRING'
: use STRING as build subdirectory for the package.
Some packages' build system is nested in some subdirectory. Use this option to specify a directory to move to before configuring and building the package.
This is a real example with tcl source package, that has a unix subdirectory that contains the configure script. Trying to build it normally gives:

```
$ sourceinstall --add tcl8.4.11-src.tar.gz -C " --enable-shared --disable-threads --enable-symbols"
sourceinstall: warning in 'Configuration': A configure script for this package has not been found in the top source directory. However, configure scripts have been found in subdirectories

unix/configure
win/configure
tools/configure

Use the '--subdir' option to specify a build subdirectory containing one of these configure scripts
# configure script not available
sourceinstall: warning in 'Configuration': configure script not available. Default prefix and configure options will be ignored.
# compile software
sourceinstall: error in 'Install package': Could not compile the code.
```

In this case, the subdirectory to indicate is of course the unix subdirectory:

```
$ sourceinstall --add tcl8.4.11-src.tar.gz -C " --enable-shared" -D unix
```

'-f, --force'
: force execution of the action even when not recommended (Not Implemented Yet).
Currently this option has absolutely no effect.

'-U, --user STRING'
: privileged user login to revert to if required [default=root] .
When running `sourceinstall` as a regular user, sometimes privileges will be needed, for example to install to a part of the filesystem that belongs to root, or even to prepare binary packages (see '--pack' action).
You can choose to use a login different than root in these cases, by specifying this option.

'-P, --pass STRING'
: privileged user password to revert to if required [default=] .
WARNING! USING THIS OPTION IN A MULTIUSER ENVIRONMENT IS VERY INSECURE!
When running `sourceinstall` as a regular user, sometimes privileges will be needed, for example to install to a part of the filesystem that belongs to root, or even to prepare binary packages (see '--pack' action).
You can use this option to specify the password to use in these cases. Note however that the program invocation, with all its parameters, could be stored in some log files, or in the system running processes list.
If you want to run `sourceinstall` from the command line in a multiuser environment, you will probably do best by running sourceinstall as the root user.

'-q, --quiet'
: be very quiet: only indispensable information will be sent to the standard output.

3.4 Preferences

Preferences are very much like options, but they are stored in a specific file that is loaded each time sourceinstall starts.

Every time a command line option and a preference clash, the command line option takes precedence.

To change the preferences you can edit your '~/.sourceinstall/sourceinstallrc' file.
Note that each user (root too!) has his own preferences file.

'MANUAL_CONFIGURE: bool'
: (Tk interface only) This preference controls whether the configuration window should be shown in the Tk interface. It has no effect if you are using the command line interface.
The 'bool' value can be 0 (do not show configure window) or 1 (do it).

'KEEP_SOURCE: bool'
: This preference controls whether the source code should be stored when adding or installing software. Of course, this preference clashes with options '--source', '--binary', and their aliases. The 'bool' value can be of course 0 or 1.

'INSTALL: bool'
: This preference controls whether the software should be installed when adding software. Of course, this preference clashes with options '--source', '--binary', and their aliases. The 'bool' value can be again 0 or 1.

'ADD_DIRECTORY: path'
: (Tk interface only) This preference controls the initial directory 'PATH' to browse when clicking Add.

'STRIP: bool'
: This preference mimics the '--strip' option.

'PREFIX: path'
: This is the default prefix to use when adding/installing software. Note that you can override this setting by providing a '--prefix' configure string (see the '--configure' option). The default value is '/usr/local', a safe choice for systemwide installs.

'SRC_COMPRESS: format'
: This is another preference that mimics an option. In this case it is the '--compression' option.

4 Usage for Beginners

4.1 The default setup

For beginners, a default setup known to work well for GNU Source Installer is the following:

1. GNU Source Installer is installed system-wide as the root user in a different prefix than your OS distribution. For example '`/usr/local`'.
2. You login normally as you always do with your user name, then start the GNU Source Installer using command `sourceinstall`, which starts the graphical interface. You do NOT login as root. At the appropriate times, during installation, you will be asked for the root password if necessary.
3. The default prefix in your preferences is '`/usr/local`', and thus you install your source packages in '`/usr/local`'
4. You are the only one on the system that installs source packages, and always do that logging in as the same user.
5. When you need to remove or upgrade `sourceinstall`, login as root and run the program. You will see GNU Source Installer listed in the root user's packages and will be able to manage it.

4.2 Looking for the right package

First of all, think about a software you want. It is highly probable that such a software package is available under a Free license somewhere.

You can search by simply using a web search engine.

Tip: add GPL or another Free license name to your search, so you are sure to find real Free Software, and not freeware, shareware or whatelse. You can also try the term "Open Source".

The Free Software and Open Source movements have different goals, but search engines tend to find pages with the term "free" as in no-cost, while free software is about freedom.

`http://www.gnu.org/philosophy/free-software-for-freedom.html` explains the relationship further.

You can choose another road, and use a Directory instead. Good places to start are the Free Software Directory and Savannah (home to the `sourceinstall` project development). Other good places to search are Freshmeat and Sourceforge, although you will find a lot of not really Free software there too.

Once you have found an interesting software, look for a SOURCE download (.tar.gz, .tar.bz2, ..) Proceed with the download, and mark where the file will be downloaded.

4.3 Adding a new source package

Once you have a new shiny source package, it is time to add it from the Source Installer. Run `sourceinstall`, then press the `Add` button.

In the `Add` dialog that appears, you can choose `Browse` to locate the package, and finally choose `Ok` to proceed. Let the other checkboxes be with their default values.

If everything runs smooth and the package has been built with the autotools, you will be presented with a configuration window, where all package options can be tweaked prior to installation.

If you have no idea about what those options mean, at least take a look at the option descriptions. You can then try 'Auto' to go on with the defaults.

The option '`--prefix`' will be highlighted. This is because it's a very useful and important option, that lets you specify where your install tree should start.

When you are satisfied with the options, choose `Ok` and wait for the software to be configured and compiled.

If no problems occur, you will be eventually asked for the root password (if needed), and then you will be informed about the result of the install operation.

4.4 Changing the Preferences

The default prefix to use for your installs can be changed, like other options, in the `Preferences` from the `Edit` menu, and it is initially set to '`/usr/local`'.

Here are the preferences you can change and their description:

"Default installation choices: Manual configuration, Install, Keep Source" These are the default values for the checkboxes when you trigger the `Add` action. Beginners should keep all those selected.

"Manual configuration" means that you will be able to see the software configuration window. It will get you aquainted with the common options supported by the packages, so it is recommended to keep this selected.

You can always choose `Auto` in the configuration window to stick with the defaults.

"Install" means that when you add new packages, they will be installed. Most beginners would want this.

"Keep Source" means that the configured source code is compressed, archived and stored for later use. This does waste some space, but ensures a cleaner uninstall process, and can provide a future easy reinstallation.

"Strip binaries (not recommended)": this option should be off. It can cause a lot of trouble if you don't know what you are doing. It involves removing symbolic information from the installed programs.

"Default install prefix": this is the default prefix to use when installing software packages. Programs and data will generally be installed in a subtree of the specified directory. The default value is '`/usr/local`' and is a good one for system-wide installs.

"Src compression": this is the compression format to use when archiving source packages. By default it is .bz2 (which provides very space-efficient compression), but if you have plenty of space in your disk and prefer quicker installs and uninstalls you can change it to .gz

4.5 Querying package information

Clicking on the package will show all available information on that package, and will activate the actions for the installed package: `Remove` and `Reinstall`. This will also trigger a quick check to ensure that the package has all its needed files in place.

4.6 Removing a package

To remove a package, select it from the list and click the `Remove` button. When you `Remove` a package, you can decide to uninstall the package but keep it in compressed source form. This way, should you decide to install again later, you have the already configured source, and only need to select it from the list and choose `Install`.

These instructions should get you started. Read on if you want to know more.

5 Usage for the Experienced

A package processed by sourceinstall can exist in three forms:

1. installed `+` archived configured source
2. installed only
3. archived configured source only

For example, if you are short on space, and you are installing a conforming package (so you get a list of installed files in the package details), you can decide to remove the archived source (losing all the advantages though) to free up some space. Alternatively, you can avoid to store it in the first place when you perform the `Add` action.

On the countrary, you might think that you do not require a certain installed program right now, so you select the `Remove` action for that package, but remove only the installation and not the source, so you can quickly reinstall should you require the software again in the future. Your configuration will be preserved, you will not need to pass through the configuration window anymore if you were fine with the last installation.

5.1 Consistency checks

Another service that `sourceinstall` offers is a simple set of consistency checks for existing installations.
In the Tk interface, selecting a package from the list at any time will show all available information about the installed package, and a check will be performed to see if the install looks ok. If some of the files required by the program are missing, you will find a notice and all the missing files will be highlighted and marked with asterisks (*).
The same thing can be obtained from the command line interface using the '`--check`' action.

At this point you can correct the problem by restoring the missing files yourself (for example, you might have accidentally moved them for arcane reasons), or just reinstall the package, using the `Reinstall` button of the Tk interface, or the '`--install`' action of the command line one.

Additionally, during the uninstallation cross-checks will be performed between make uninstall results and internal package information; only independent files (that is, files that are not being claimed by other packages) are proposed for removal (Tk interface), or directly removed (command line interface).
This works better if you avoid installing non-conforming packages, because Source Installer will not be able to know which files a non-conforming package claims.

5.2 Conforming packages

It is recommended to install only conforming packages using Source Installer. Conforming packages offer a configure script that generates a Makefile, and the Makefile honors the common install targets and environment variables. Generally, packages built (correctly) with the GNU autotools result as conforming packages, and the autotools are also expecially supported: only configure scripts generated with autoconf get the nice configuration window in the Tk interface. However, there are also other tools that developers can use which are capable of producing a configure script and a Makefile. Even hand-written configure scripts and Makefiles are ok, as long as they honor the install targets and the environment variables.

Conforming packages get better uninstallation, better checks, more information in the Package information window. A single non-conforming package can make uninstallation checks degrade. This is because the program can not detect which files a non-conforming package claims.

For these reasons, you will be warned when installing a non-conforming package.

5.3 Other package management systems

It is highy probable that you will have many different tools that provide package management. For example, if you are running a GNU/Linux distribution, you probably have your distribution-specific way to handle binary (or even source) packages. What I suggest here is to make a clean separation between your distribution-provided packages (along with any additional packages installed using your distribution-specific tools), and the source packages installed using the distribution-independent GNU Source Installer. One good way to obtain this, is to use different prefixes for each package management system you use. Suppose your distribution-handled packages are in '`/usr`', then your source packages managed by GNU Source Installer can be prefixed using '`/usr/local`' (this is the initial value). This way you ensure that file dependency checks are not tainted by other packages managed by other tools.

5.4 Users

GNU Source Installer configuration and packages always refer to the particular user that runs it. What follows is a list of possible setups.

As the first example, suppose user Pip wants to install his private packages. He chooses to install in '`/home/pip/usr`', at the same time allowing user Merlin to install his own packages in '`/home/merlin/usr`'. Of course, in this case no one steps on anybody's toes and everything runs smooth.

As the second example, user Merlin is the system administrator, and personally deals with all system-wide package installations. Thus, he logs in using his `merlin` account, then runs `sourceinstall` and uses the default prefix value, which is '`/usr/local`' to install new programs. When requested by the system, he is asked for the root password. This is ideal for one-user systems.

As a third example, suppose both Pip and Merlin deal with system administration. They get along well, and both deal with system-wide package installations. Thus, they decide to both install packages logging in using the privileged `root` account, and then install using `sourceinstall`. They will both see the same packages, because they are logging in as the same user (`root`). However, only one of the two admins will be able to install software at the exact same time, because to prevent corruption of package data each user is entitled to a single running instance of GNU Source Installer. When the program is already in use, `sourceinstall` refuses to start and explains the error.

As a last example, here's what not to do. Imagine both Pip and Merlin deal with system administration, like before, but since they don't read the docs, they login using their regular accounts, and perform installs using the default system-wide prefix, '`/usr/local`', providing the root password when needed. Even if they install at different times this is a bad thing to do: they will not be aware of each other's moves, file dependency tracking will be far less precise, and uninstallation crosschecks will degrade.

5.5 Files and Directories

If you want to dwell on the internals of sourceinstall, this is an interesting chapter. We will look at all files and directories that together make sourceinstall work.

The `sourceinstall` executable is in fact a symbolic link to the implementation in use. For example:

```
$ ls -l /usr/bin/sourceinstall*
lrwxrwxrwx  1 root root     11 Jun  3 03:15 sourceinstall -> sourceinstall.tcl
-rwxr-xr-x  1 root root  87019 Jun  3 03:15 sourceinstall.tcl
```

Currently there is only a tcl implementation, but in the future this system will be used to make different implementations of `sourceinstall` coesist.

The per user configuration directory is another story:

each user that runs `sourceinstall` gets a .sourceinstall directory created in his `HOME`. This is for example a directory tree:

```
$ tree ~/.sourceinstall
/home/claudio/.sourceinstall
|-- build
|-- install-destdir
|-- packages
|   |-- a-renet-1.1.0rc5
|   |-- cmdftp-0.9.2
|   `-- libmikmod-3.2.0-beta2
|-- src
|   |-- a-renet-1.1.0rc5.tar.bz2
|   |-- cmdftp-0.9.2.tar.bz2
|   `-- libmikmod-3.2.0-beta2.tar.bz2
`-- sourceinstallrc
```

If sourceinstall were running, you would see another file, '`~/.sourceinstall/.sourceinstall_lock`', containing the process id of the running sourceinstall process.

In this case there are only three packages installed. Each package has a file name entry with the same name in the '`packages`' directory. That file contains all information on that package.

The source for all three packaged has been archived in the '`src`' directory.

The '`install-destdir`' directory is used during the test installation, and then it is cleared.

The '`build`' directory is used only while building packages, and then it is cleared.

Do not store anything in these directories yourself, because they will be regularly emptied.

The 'sourceinstallrc' file contains the current user preferences. These are the same options that can be tweaked from the Preferences in the Edit Menu using the Tk interface.

5.6 Export functions

sourceinstall contains now some experimental export functions, to offer more interface to other programs (and some user functionalities as well).

Information about a single package (or about all the packages) can be exported through the *Export package information to XML* and *Export package information to plaintext* features. The XML involved is a simple XML 1.0 format (look at the XML output for the external public DTD address) and is more machine-oriented, while the plaintext is more user-oriented.

It is also possible to extract just the list of the files claimed by the installed package, in a plaintext, newline-separated list using the *Export installed files list* action.

A whole different story is the *Export as binary package* functionality. Information that follows is very unstable and should not be relied upon for the long term. This is a simple function that builds a .tar.gz archive out of the files, directories and links claimed by an installed package. Original file permissions and ownership will be preserved. It should be easy to build OS-specific binary packages out of these simple binary tarballs. However, also consider using the export installed files list functionality instead. If you expected more binary package building functionalities, please remember that this program is source-centered and OS-neutral.

6 FAQ

This is a small section in which I try to address common questions.

1. Q: GNU Source Installer does not install! What should I do?

   A: There are dedicated topics in this manual.

   See Chapter 1 [Installing the Installer], page 1.
   See Chapter 1 [Troubleshooting Installation], page 1.

2. Q: I have found a bug in GNU Source Installer! What should I do?

   A: By all means, do report it. You will get the problem solved, and the software will become better if users report bugs. See Chapter 7 [Reporting Bugs], page 24.

3. Q: sourceinstall coud not install Package [:xyz:]. What can I do?

   A: If the package does not provide a build system compatible with `sourceinstall`, you will get an error message, stating that the program could not compile the code. To get the package supported, you have at least two choices.

   If you are an experienced user, you should contact the original author or maintainer of that package, explaining that you are trying to install it using a new install tool (GNU Source Installer), but it does not work because it does not provide a build system compatible with the idiom:

   ```
   ./configure
   make
   make install
   ```

   Then provide a quick solution for the author, so he will not need any effort to accept the change. If you are not sure, rely on the `sourceinstall` people to do this for you.

   If you are a little less experienced and do not know how to help the author of the package, just contact the `sourceinstall` mailing list at `bug-sourceinstall@gnu.org`. We will contact the package maintainer, and help him make that package work with `sourceinstall`.

4. Q: Why doesn't sourceinstall use Checkinstall?

   A: Checkinstall is a nice program that tracks a source installation using Installwatch, which is Copyright 1998 by Pancrazio 'Ezio' de Mauro, and is now part of the Checkinstall distribution. Installwatch is a low-level tool that intercepts calls to file functions in the dynamically linked glibc that alter the file system during 'make install' (or another installation command). Then, Checkinstall builds binary Slackware, RPM or Debian packages based on that information.

   Now I find that the DESTDIR and INSTALL_ROOT (its old, deprecated replacement) way used in `sourceinstall` is more clean and portable (although slower) than the low level installwatch approach, and is an incentive for developers to correctly support DESTDIR in their custom Makefiles or in their automake hooks. `sourceinstall` detects if the build system supports DESTDIR or the old INSTALL_ROOT variables, and uses them for the installation tracking.

   As for GNU/Linux distribution-specific binary package building features, GNU Source Installer is again an OS neutral program, so it is not its job to build them. However, a functionality to build simple, neutral binary packages in the form of a tarball archive is available. From that package, it should be straightforward to build your desired distribution-specific package, but again it is not the point of this program.

5. Q: Does sourceinstall build [:your format:] binary packages ?

   A: (see question above) It has been recently reported that `sourceinstall` simple binary packages happen to be handled correctly by the Slackware GNU/Linux tools. That is simply because both use plain tarballs which contain the needed files with the original owners and permissions. Remember that building binary packages is not the point of this program (but use this as you see fit of course).

6. Q: There's Gentoo and Portage already. Isn't sourceinstall pointless ?

   A: Surprisingly, I got many (well some) private mails stating that there's already the Gentoo GNU/Linux distribution and its Portage system, so `sourceinstall` was supposedly pointless. Again, this software is an installer and source package management tool, targeted at modern Unix systems (these include, but do not end with GNU/Linux). Of course, `sourceinstall` is not a GNU/Linux distribution and not a package repository.

7. Q: Why are you using Tk? Why don't you use gtk|Qt|[:yourtoolkit:] ?

   A: I do not have any real preference of one toolkit over another. They can all get the job done. However I really wished to use the GNU ToolKit at the beginning (gtk), because it's the GNU ToolKit (talk about tautology). There is one problem: dependencies and dependencies' size. I wanted to provide something that you could install from a bare-bones Unix + X11 system requiring nothing else.

   With the current tcl/tk/expect implementation I can provide, in 7 MB, a fullpack alternative shell-archive release of `sourceinstall`, which includes and autobuilds all its dependencies if necessary. The same thing would not be possible with gtk, which is bigger and less straightforward to autobuild.

   If you want a [:yourtoolkit:] interface to the program badly, you can implement it on top of sourceinstall command line interface (but contact the author so you can get all the help and possibly tweaks to the young interface that you need).

8. Q: Why are you using Tcl? Why don't you use [:yourlanghere:] ?

   A: Tcl seemed the natural choice having to deal with both Tk and Expect, and it offered simpler Unix portability. However this could change. I am not happy with some of the tcl decisions, and some bugs in the implementation do not help either. If the whole project language changes, it will change to C.

9. Q: I do not care about cross Unix portability, dependencies, building the whole system from scratch or anything like this. Is there an alternative, specific for my already working GNU/Linux desktop system ?

   A: There are projects with somewhat similar goals that I have been made aware of:

   Kconfigure, QT based, for KDE: `http://kconfigure.sourceforge.net/`

   Easinstaller, Ruby and Fox based: `http://easinstaller.sourceforge.net/`

   GPaco, Gtk based, for GNU/Linux and Solaris: `http://paco.sourceforge.net/`

   I know little about them save their name so you should investigate them yourself.

10. Q: The manual is not comprensive enough! It does not say anything about [:X:]

    A: The project is still very young, as is the documentation. It will be a lot better; you can speed up things by reporting what exactly is missing, and possibly providing yourself if you have some time.

11. Q: This program does not deal with dependencies at all!

    A: This manual will be integrated with a detailed explanation on how to deal with dependencies. With some patience, I think almost everyone can learn to live with them.

12. Q: I have an older version of sourceinstall, how do I upgrade to the current version?

    A: Starting from sourceinstall-0.4, this has been made easier. Just download the regular new version (for example, sourceinstall-0.5.tar.gz), then run sourceinstall, and choose "Add"; select that new sourceinstall package (sourceinstall-0.5.tar.gz), and confirm with Ok. All your installed packages and preferences will not be harmed.

13. Q: The right click popup menu does not work (anymore)!

    A: The Tk popup menus (and consequently sourceinstall prior to v-0.5) were not sticky. This means that in order to keep the menu visible, one had to keep the button always pressed. This has been fixed very recently in a development version of Tk.

    In order to offer a more usable popup menu, starting from 0.5 I use a workaround that involves showing the popup on right click button RELEASE. In short: starting with sourceinstall 0.5, click and release the right button on a list entry, and the popup will appear.

14. Q: I want to contribute to the project in some way, where do I start?

    A: Start with the Bugs and Task list in the Savannah sourceinstall project page: `http://savannah.gnu.org/projects/sourceinstall/` . Even if you are not a programmer, the task list can contain very relevant work that does not involve programming or reading code at all.

    If you have an interesting idea to share, by all means do so. You can contact the mailing list `bug-sourceinstall@gnu.org`, or if you prefer you can contact the author directly at `claudio@gnu.org`. As long as your mail is polite enough, you will not be judged by me in any way by the ideas you express, even if I do not agree with you.

15. Q: I want to join the project! Please add me to the project members list.

    A: Great, but please do something for the project. If you make regular contributions, and understand the project goals, you will be added to the sourceinstall project.

7 Reporting Bugs

Email bug reports to `bug-sourceinstall@gnu.org`, trying to be as clear and precise as possible. This means that you should provide all useful information that could help to identify the problem, and a detailed way to reproduce it. A good starting point is to specify your OS name and version. If you have no idea about what OS you have, try:

```
$ uname -a
```

Also, if you can please specify your tcl, tk, and expect versions.
If your problem regards GNU Source Installer installation, more information is needed: please read the sections "Installing the Installer" and "Troubleshooting Installation".

8 Copying

GNU Source Installer (`sourceinstall`) is Copyright (c) 2005 Claudio Fontana and is licensed under the GNU General Public License (GPL). This is a license that grants and protects freedoms. `sourceinstall` is thus Free Software. There is an exception in the m4/tcl.m4 file, which is Copyright (c) 1999-2000 Ajuba Solution, Copyright (c) 2002-2005 ActiveState Corporation. The m4/tcl.m4 file is licensed under a “revised BSD” license. See m4/license.terms. It has been slightly modified for GNU Source Installer.

Please note that “Free” in “Free Software” refers to liberty, not price. Think of “free speech” rather than “free beer”. The exact and legally binding distribution terms are spelled out below; in short, you have the right (freedom) to run and change sourceinstall and distribute it to other people, and even—if you want—charge money for doing either. The important restriction is that you have to grant your recipients the same rights and impose the same restrictions.

This method of licensing software is also known as *open source* because, among other things, it makes sure that all recipients will receive the source code along with the program, and be able to improve it. The GNU project prefers the term “free software” for reasons outlined at `http://www.gnu.org/philosophy/free-software-for-freedom.html`.

The exact license terms are defined by this paragraph and the GNU General Public License it refers to:

In addition to this, this manual is free in the same sense:

The full texts of the GNU General Public License and of the GNU Free Documentation License are available below.

8.1 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

1. 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.

2. 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.

3. 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.

4. 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.

5. 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.

6. 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.

7. 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.
8. 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.
9. 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.
10. 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.

11. 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

12. 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.
13. 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 an idea of what it does.
Copyright (C) 20yy  name of author

This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the
Free Software Foundation, Inc.,
51 Franklin Street - Fifth Floor, Boston, MA 02110-1301, USA.
```

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

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

```
Gnomovision version 69, Copyright (C) 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.

8.2 GNU Free Documentation License

Version 1.2, November 2002

0. PREAMBLE

   The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

   This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

   We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

   A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

   A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission. B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement. C. State on the Title page the name of the publisher of the Modified Version, as the publisher. D. Preserve all the copyright notices of the Document. E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices. F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below. G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice. H. Include an unaltered copy of this License. I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence. J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission. K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein. L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles. M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version. N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section. O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties–for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

   Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

   If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

   You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document 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.

10. FUTURE REVISIONS OF THIS LICENSE

   The Free Software Foundation may publish new, revised versions of the GNU Free Documentation 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. See http://www.gnu.org/copyleft/.

   Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

```
To use this License in a document you have written, include a copy of
```

```
the License in the document and put the following copyright and
license notices just after the title page:

    Copyright (c)  YEAR  YOUR NAME.
    Permission is granted to copy, distribute and/or modify this document
    under the terms of the GNU Free Documentation License, Version 1.2
    or any later version published by the Free Software Foundation;
    with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
    A copy of the license is included in the section entitled ``GNU
    Free Documentation License''.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts,
replace the ``with...Texts.'' line with this:

    with the Invariant Sections being LIST THEIR TITLES, with the
    Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other
combination of the three, merge those two alternatives to suit the
situation.

If your document contains nontrivial examples of program code, we
recommend releasing these examples in parallel under your choice of
free software license, such as the GNU General Public License,
to permit their use in free software.
```

Beginners Concept Index

This is an index of the most basic topics and definitions used in this manual. If you are not sure about a notion or term used in the manual, you can try looking for it here.

Concept Index

This is a list of all the relevant topics in the `sourceinstall` manual.

This is an alphabetical list of all `sourceinstall` functionalities, commands, command-line actions and options, and relevant environment variables.

Table of Contents