

Licenseutils

License Utilities
for version 0.0.8pre

Ben Asselstine

This manual documents version 0.0.8pre of Licenseutils.

Copyright © 2013 Ben Asselstine

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

1 Introduction

Boilerplate is that necessary bit of text at the top of source code files that conveys the copyright holders and license of the file. It's not always identical from file to file, but it certainly has a standard look and feel to it.

Programming languages generally support copyright and license notices through their commenting facility. For example, the C language has a different commenting style than haskell: `/* foo */` vs `{- bar -}`

Free software depends on copyright and license notices being present because individual files have a way of propagating independently from their parent project.. Adding these notices can be time-consuming even for small projects.

licenseutils consists of a program `licensing`, and a shell called `lu-sh`. Together they provide a set of commands to quickly and easily modifying files to include a standard-looking free software boilerplate for various different programming languages.

licenseutils uses the GNU Source Highlight library to detect comments. www.gnu.org/software/src-highlight/

If you're wondering how to add license notices to your files, here is a quick-start example:

```
$ licensing new-boilerplate
$ licensing choose gpl
$ licensing choose c
$ licensing copyright My Name
$ licensing apply *.c
```

1.1 licensing: Licensing utility program.

The `licensing` program has various commands. To run them, type:

```
$ licensing COMMAND
```

Some commands involve the concept of a current working boilerplate, and some do not. The current working boilerplate is a notional block of text representing the boilerplate that the user wants to add to a file. It lets the user easily change the commenting style, or copyright holders before applying a similar boilerplate to another kind of file.

1.1.1 Commands that create the current working boilerplate

There are commands for creating a current working boilerplate: (see Chapter 3 [Creating boilerplate], page 8)

`new-boilerplate`

Resets the current working boilerplate. Any previously selected licenses, commenting styles, or copyright holders are thrown out. See Section 3.1 [new-boilerplate invocation], page 8.

`choose` Adds or changes the license notice or commenting style for the current working boilerplate. See Section 3.2 [choose invocation], page 8.

`copyright`

Adds or removes a copyright notice to the current working boilerplate. See Section 3.3 [copyright invocation], page 9.

- top-line** Add a beginning line to the current working boilerplate to describe what the file is for. See Section 3.4 [top invocation], page 11.
- project** Adds a project-identifying line to the current working boilerplate. See Section 3.5 [project invocation], page 11.
- extra** Adds some extra text to the current working boilerplate See Section 3.6 [extra invocation], page 12.

1.1.2 Commands that write the current working boilerplate to files

The current working boilerplate can be subsequently written to files: (see Chapter 4 [Writing boilerplate], page 13)

- apply** Writes the current working boilerplate to source code files. See Section 4.1 [apply invocation], page 13.
- png-apply** Writes the current working boilerplate to .png image files. See Section 4.2 [png-apply invocation], page 14.
- notice** Creates and writes a simple boilerplate to a source code file. This command does not reference the current working boilerplate. See Section 4.4 [notice invocation], page 15.
- prepend** Adds some arbitrary text to the start of a file. This command does not reference the current working boilerplate. See Section 4.3 [prepend invocation], page 14.

1.1.3 Commands that get boilerplate from files

There are a few commands for showing boilerplate in files: (see Chapter 2 [Scanning for boilerplate], page 5)

- boilerplate** Show the boilerplate in source code files. This command can also remove the boilerplate from a file. See Section 2.1 [boilerplate invocation], page 5.
- cbb** Counts the number of distinct boilerplate blocks in source code files. See Section 2.2 [cbb invocation], page 6.
- png-boilerplate** Show the comment in .png image files. This command can also remove the comment from a .png file. See Section 2.3 [png-boilerplate invocation], page 7.

1.1.4 Commands that display license notices

There are commands that display licenses and their notices: (see Chapter 6 [License commands], page 17)

- gpl** Shows various versions of the GNU General Public License notice, or optionally the texts of the full license. See Section 6.1 [gpl invocation], page 17.
- lgpl** Shows various versions of the GNU Lesser General Public License notice, or optionally the texts of the full licenses. See Section 6.2 [lgpl invocation], page 18.

- agpl** Shows the GNU Affero General Public License notice, or optionally the texts of the full licenses. See Section 6.3 [agpl invocation], page 18.
- fdl** Shows various versions of the GNU Free Documentation License notice, or optionally the text of the full licenses. See Section 6.4 [fdl invocation], page 18.
- all-permissive** Shows the GNU All-Permissive license. See Section 6.5 [all-permissive invocation], page 18.
- fsf-permissive** Shows the FSF Permissive license. See Section 6.6 [fsf-permissive invocation], page 19.
- bsd** Shows various versions of the Berkeley Software Distribution license. See Section 6.7 [bsd invocation], page 19.
- apache** Shows the Apache license notice or optionally the full license text. See Section 6.8 [apache invocation], page 19.
- mit** Shows the Massachusetts Institute of Technology license. Also known as the X11 License. See Section 6.9 [mit invocation], page 19.
- isc** Shows the Internet Systems Consortium license. Also known as the OpenBSD License. See Section 6.10 [isc invocation], page 19.
- artistic** Shows the Artistic license notice. See Section 6.11 [artistic invocation], page 19.
- epl** Shows the Eclipse Public license notice. See Section 6.12 [epl invocation], page 20.
- mpl** Shows the Mozilla Public license notice. See Section 6.13 [mpl invocation], page 20.
- zlib** Shows the ZLib license. See Section 6.14 [zlib invocation], page 20.

1.1.5 Other commands that display information

There are commands that operate on commenting styles: (see Chapter 5 [Working with comments], page 16)

- comment** Creates a comment block in a commenting style particular to a programming language. See Section 5.1 [comment invocation], page 16.

And finally there are a few commands that are informational in nature (see Chapter 7 [Informational commands], page 21, that display various bits of information:

- preview** Shows the current working boilerplate. See Section 7.1 [preview invocation], page 21.
- welcome** Shows the welcome message. See Section 7.2 [welcome invocation], page 21.
- warranty** Shows the warranty message. See Section 7.3 [warranty invocation], page 22.
- help** Shows help on all of these commands. See Section 7.4 [help invocation], page 22.

1.1.6 Starting the lu-sh shell

When a command is not given as an argument to `licensing`, the interactive `lu-sh` shell is started. The `--quiet` option prevents the welcome message from being displayed in the `lu-sh` shell.

The `lu-sh` shell is an extended bash shell. The initialization file for `lu-sh` is automatically generated, and lives in `~/.lu-shrc`.

The program state (e.g. the current working boilerplate) is kept in `~/.licensutils/`.

All of the `licensing` commands work in the `lu-sh` shell without a `licensing` command prefixed to it.

For example:

```
#!/usr/bin/env lu-sh
welcome
exit 0
```

This `lu-sh` script is equivalent to running the command: `licensing welcome`. Although this example shows a `lu-sh` script, the shell is most often used interactively.

2 Scanning for boilerplate

These commands read boilerplate from files, possibly transforming it in some way.

2.1 boilerplate: Show or remove boilerplate in source files

The boilerplate command shows the comments at the top of source code files. The most common execution of this command is:

```
$ licensing boilerplate foo.c
```

The boilerplate command will automatically determine what kind of comment-style is being employed at the beginning of the source code file. This automatic detection has some drawbacks, so various commenting-style options are provided to force the detection of comments in a particular commenting style. See Chapter 9 [Common Commenting-style options], page 24.

2.1.1 Rules For Automatic-detection of Comments

Automatic detection of commenting-styles is a heuristics that checks for commenting-styles in a certain order, sometimes avoids particular file extensions, and sometimes only operates on certain supported file extensions. The order for automatic detection is: C, C++, javascript, shell, scheme, texinfo, m4, haskell, groff, gettext, fortran, pascal.

C style comments are featured in other languages: Java, PHP, Javascript, Go, and many more. Shell style commenting is used in python, make, perl, and more.

When auto-detecting shell style comments will not be found in files with extensions: `‘.c .h .cpp .hpp .hh .cc .m4 .ac .po .pot’`. M4 comments can only be found in files with extensions: `‘.m4 .ac’`. Fortran comments can only be found in files with extensions: `‘.f .for .f90 .f95’`. These strategies fail when files are passed via the standard input: e.g. when the file is `-`.

2.1.2 Removing boilerplate

The boilerplate command can also remove boilerplate from one or more source code files by specifying the `--remove` option. For example:

```
$ licensing boilerplate --remove foo.c
```

By default the commenting style is auto-detected, but each of the commenting style options can be used to remove comment blocks of a specific commenting style. Particular comment blocks can be removed with the `--blocks` option. The `--force` option is used to remove comment blocks that contain copyright notices. When removing boilerplate, a backup file (`‘.bak’`) is usually made. The `--no-backup` option is provided to prevent this behaviour. When removing boilerplate from the standard input it cannot be backed up, and the resulting file is simply displayed.

The `--quiet` option prevents error messaging from being displayed when a boilerplate is not found, or an invalid comment-block is referred to with the `--blocks` option, or when removing a comment-block that contains a copyright notice and the `--force` option has not been given.

2.1.3 Showing or removing specific blocks

Sometimes the start of a source code file can have more than one comment-block. Block-style comments denote a comment-block, and whitespace separates blocks of whole-line comments.

The `--blocks` option can show or remove one or more comment-blocks.

For example `'licensing boilerplate -b1 foo.c'` shows the first comment-block. And `'licensing boilerplate -b1,3 foo.c'` shows the first and third comment-blocks. `'licensing boilerplate --remove --blocks=1-3,5 foo.c'` removes the first three comment-blocks, and the fifth comment-block too.

Whitespace between comment-blocks is not shown when the `--blocks` option is employed.

To find out how many comment-blocks there are in a file, use the `cbb` command. See Section 2.2 [cbb invocation], page 6, for more information.

2.1.4 Dealing with source code files that are scripts

When the source code file begins with a `'#!'` line, the boilerplate command does not show or remove that line.

2.2 cbb: Count the boilerplate blocks in source files

The command `cbb` stands for "count comment blocks". The purpose of this command is to display the size of the boilerplate in the given files. Here is how the command is normally executed:

```
$ licensing cbb foo.c
2 17 foo.c
```

This example shows that there are 2 comment-blocks at the beginning of `foo.c`, having 17 lines in total.

By default the `cbb` command automatically detects comments, but any of the commenting-style options can be used to detect a particular commenting-style instead. See Chapter 9 [Common Commenting-style options], page 24, for more information.

The `--lines` option causes `cbb` to display only the number of lines in the boilerplate, and the `--blocks` option displays only the number of blocks.

The `cbb` command can assist in iterating through the various comment-blocks in boilerplate.

```
#!/bin/bash
blocks='licensing cbb foo.c | cut -f1 -d' ''
for (( b=1; b<=$blocks; b++ )); do
    licensing boilerplate -b$b foo.c
done
```

When the input file is `-`, it is read from the standard input.

2.3 png-boilerplate: Show or remove the boilerplate in .png files

Sometimes PNG image files contain copyright and license notices. Authors put these notices in comments section so that when the file propagates, the license propagates too. The `png-boilerplate` command is for showing or removing comments in PNG files. It is normally executed in this fashion:

```
$ licensing png-boilerplate foo.png
```

This example will result in the comment section of the png file being displayed. The PNG file format allows for many named text sections in the file. The `png-boilerplate` command only operates on the first text section named "Comment".

2.3.1 Removing comments from a PNG file

To remove the boilerplate from a PNG file, use the `--remove` option:

```
$ licensing png-boilerplate --remove foo.png
```

This command will modify the first text section named "Comment" in the PNG file `foo.png` so that it contains an empty string. This is slightly different than removing the text section entirely.

When removing comments, a `.bak` backup file is created by default. To prevent this behaviour use the `--no-backup` option.

3 Creating boilerplate

These commands modify the current working boilerplate in some way. It is natural to run the `preview` command after running one of these commands to see how the current working boilerplate looks.

3.1 new-boilerplate: Reset the current working boilerplate

The `new-boilerplate` command throws out the current working boilerplate. The command is usually executed like so:

```
$ licensing new-boilerplate
```

Any previously selected licenses or commenting styles are forgotten. Any previously specified copyright holders are also forgotten. Lastly, if a project identifier was included, or a top line or an extra line was specified, it is also forgotten.

The `new-boilerplate` command is equivalent to:

```
licensing choose no-license no-style --quiet
licensing copyright --remove --quiet
licensing project --remove --quiet
licensing top --remove --quiet
licensing extra --remove --quiet
echo "Removed." >&2
```

The `--quiet` option prevents the final ‘Removed.’ message from being displayed.

3.2 choose: Pick license and comment style for boilerplate

The `choose` command selects a license notice and commenting style for the current working boilerplate. The most common execution of this command is:

```
$ licensing choose gpl c
Selected.
```

This example selects the latest version of the GNU General Public License and the C commenting style. It is equivalent to:

```
$ licensing choose c
Selected.
$ licensing choose gplv3+
Selected.
```

To perform dual-licensing, more than one license can be selected. For example:

```
$ licensing choose gpl apache c
Selected.
```

When more than one commenting style is given to the `choose` command, the final commenting style is the one that is selected.

Tab completion will display the available commenting styles and licenses. Alternatively, `choose --help` will suffice.

Although all of the available licenses are compatible with the GPL, some of them are not recommended. To select one of the unrecommended licenses, the `choose` command must be called with the `--force` option.

When `choose` is executed in the `lu-sh` shell, the prompt changes to reflect the selected licenses and commenting style.

The `--quiet` option prevents the final ‘Selected.’ message from appearing.

The `choose` command requires internet connectivity to download the licenses from the `www.fsf.org` or `www.gnu.org`.

3.2.1 Subsequent calls of the `choose` command

Subsequent calls to the `choose` command will override previous calls. After the current working boilerplate has been applied to some source files, it is often useful to `choose` a different commenting style before applying it to another file (e.g. all other aspects of the current working boilerplate are left unchanged except for the commenting style).

3.3 `copyright`: Add a copyright notice to the boilerplate

The `copyright` command adds or removes copyright notices to or from the current working boilerplate. When the `copyright` command is called without any arguments, the copyright notices in the current working boilerplate are displayed.

The command is most commonly used to add copyright holders in this way:

```
$ licensing copyright Yoyodyne, Inc. 2001
Copyright (C) 2001 Yoyodyne, Inc.
Added.
```

Years can be given in a comma-separated list, and with a range separated by a hyphen. For example:

```
$ licensing copyright Yoyodyne, Inc. 2001,2003-2005
Copyright (C) 2001, 2003, 2004, 2005 Yoyodyne, Inc.
Added.
```

When the years happen to be supplied with an incorrect ordering, (e.g. 2003 before 2001) the `copyright` command will sort them properly. When a year is not supplied, the current year will be shown.

The `--abbreviate-years` option (`-a`) will abbreviate the years as expected:

```
$ licensing copyright Yoyodyne, Inc. 2001,2003-2005 -a
Copyright (C) 2001, 2003-2005 Yoyodyne, Inc.
Added.
```

More than one copyright holder can be specified in a single call to `copyright` by passing the names through the standard input, like so:

```
cat << EOF | licensing copyright 2001-8 -a
Yoyodyne, Inc.
Thomas Pynchon
EOF
Copyright (C) 2001-2008 Yoyodyne, Inc., Thomas Pynchon
Added.
```

This example separates multiple copyright holders by commas. Also notice the way that ‘2001-2008’ is originally given as ‘2001-8’.

The `--one-per-line` option dispenses with the commas and shows a single copyright holder per line. For example:

```
cat << EOF | licensing copyright 2001-8 -a --one-per-line
Yoyodyne, Inc.
Thomas Pynchon
EOF
Copyright (C) 2001-2008 Yoyodyne, Inc.
Copyright (C) 2001-2008 Thomas Pynchon
Added.
```

When the names of copyright holders can't all fit on the same 80 character line, they are moved to the following line like so:

```
cat << EOF | licensing copyright 2001-8
Yoyodyne, Inc.
Thomas Pynchon
EOF
Copyright (C) 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008 Yoyodyne, Inc.
Copyright (C) 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008 Thomas Pynchon
Added.
```

The `--quiet` option will prevent the 'Added.' message and the copyright notices in the current working boilerplate from being displayed.

There are two options for controlling the appearance of the '(C)'. The `--small-c` option will display '(c)' instead, while the `--unicode-sign` option will display '©' instead.

The `--dry-run` option simply shows the new copyright line without changing the current working boilerplate.

When the names of copyright holders have numbers in their names that look like years, the `--year (-y)` option can be used. Using this option will cause the numbers in the arguments to be treated as a name.

Long lines can sometimes be created by the `copyright` command.

3.3.1 Subsequent calls of the copyright command

Subsequent calls of the `copyright` command will add more copyright holders to the current working boilerplate. New copyright holders appear after any previously added copyright holders.

When one copyright holder has already been specified and another is added, the `copyright` command will subsequently display two lines, not one. One line is displayed for each copyright holder.

3.3.2 Removing copyright notices

Recall that to show the copyright holders in the current working boilerplate, the `copyright` command is called without any arguments:

```
$ licensing copyright
Copyright (C) 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008 Yoyodyne, Inc.
Copyright (C) 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008 Thomas Pynchon
```

To remove the second copyright holder (Thomas Pynchon), use the `--remove` option:

```
$ licensing copyright --remove=1
Copyright (C) 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008 Yoyodyne, Inc.
Removed.
```

The argument to the `--remove` option starts counting at 0, so the second line is referenced by the number 1.

3.4 top: Add a top line to the boilerplate

Boilerplate can sometimes have an initial line that gives a short description of the file. The `top` command is used to specify that "top" line of the current working boilerplate:

```
$ licensing top "frcode.c: front-compress a sorted list"
Added.
```

Subsequent calls of the `top` command will override previous calls. The idea here is to iterate through each source file, and calling `top` before apply-ing the boilerplate.

Running the `top` command without any arguments displays the top line in the current working boilerplate:

```
$ licensing top
frcode.c: front-compress a sorted list
```

The `--remove` option will remove the top line from the current working boilerplate.

The `--quiet` option prevents the 'Added.' or 'Removed.' message from being displayed.

3.5 project: Add a project-identifier to the boilerplate

Sometimes projects like to add an identifying line to their boilerplate. This ensures that the origin of the file is clear when the file propagates out to another project.

The `preview` command adds a line to the current working boilerplate that identifies the project. The new line appears after the copyright holders in the boilerplate. It is most commonly used like this:

```
$ licensutils project GNU findutils
Added.
```

When the `project` command is executed without any arguments, the line will be displayed:

```
$ licensutils project
This file is part of GNU findutils.
```

The `--belongs` option causes an alternative identifying line to be added to the current working boilerplate.

```
$ licensutils project GNU findutils --belongs
Added.
$ licensutils project
This file belongs to the GNU findutils project.
```

Subsequent calls to the `project` command override earlier calls.

The `--remove` option is provided to completely remove the project line from the current working boilerplate.

The `--quiet` option prevents the 'Added.' or 'Removed.' messages from being displayed.

3.6 extra: Add some extra text to the boilerplate

The `extra` command specifies some arbitrary text to be shown before the license notices. Perhaps the copyright holder is an organisation and it is useful to include the actual names of the authors of the file. Or perhaps it is useful to insert some explanatory text describing how the work is dual-licensed.

To specify some extra text, use the `extra` command like so:

```
$ licensing extra "Written by Thomas Pynchon <tom@yoyodyne.net>."  
Added.
```

The double quotes are added in this example so that ‘<’ and ‘>’ aren’t treated as redirection symbols by the shell.

As with `copyright`, `top`, and `project`, when the `extra` command is executed without any arguments, the current extra text is displayed:

```
$ licensing extra  
Written by Thomas Pynchon <tom@yoyodyne.net>.
```

Subsequent calls of the `extra` command override earlier calls.

The `--remove` option removes the extra text from the current working boilerplate.

To prevent the ‘Added.’ or ‘Removed.’ messages from being displayed, use the `--quiet` option.

4 Writing boilerplate

These commands write boilerplate to files.

4.1 apply: Write the working boilerplate to source files

After a current working boilerplate has been created, and it looks good in the `preview` command, it is time to actually put the boilerplate into files. `apply` is the command that is used to write the current working boilerplate to source code files. When starting a new project, it is used like so:

```
$ ls
bar.c foo.c foo.h Makefile qux.c
$ licensing apply *.ch]
apply: foo.c -> Boilerplate applied.
apply: bar.c -> Boilerplate applied.
apply: qux.c -> Boilerplate applied.
apply: foo.h -> Boilerplate applied.
```

This example adds the boilerplate to the beginning of the `.c` and `.h` files in the current directory.

The `apply` will fail to operate when the current working boilerplate lacks copyright holders or a license.

When a source file happens to be a script, the boilerplate will appear after the `'#!'` line. To place the boilerplate after any previously existing boilerplate in the source file, use the `--after` option. When the commenting-style is not specified with an option (see Chapter 9 [Common Commenting-style options], page 24, it is automatically detected. For example:

```
$ cat foo.sh
#!/bin/bash
# hello world
echo hello world
$ licensing choose all-permissive shell --quiet
$ licensing copyright Yoyodyne, Inc. 2001 --quiet
$
$ licensing apply foo.sh --after --shell-style --no-backup
apply: foo.sh -> Boilerplate applied.
$ cat foo.sh
#!/bin/bash
# hello world
# Copyright (C) 2001 Yoyodyne, Inc.
#
# Copying and distribution of this file, with or without modification,
# are permitted in any medium without royalty provided the copyright
# notice and this notice are preserved. This file is offered as-is,
# without any warranty.
echo hello world
```

This example puts the current working boilerplate after the shell boilerplate in `foo.sh`.

By default a backup `.bak` file is created; to prevent this behaviour use the `--no-backup` option.

The `--quiet` option prevents the ‘Boilerplate applied.’ messages from being displayed.

Subsequent calls to the `apply` command are cumulative. If a mistake has been made, it can be removed with the ‘`boilerplate --remove`’ command.

The `apply` command uses the `preview` command to generate the current working boilerplate.

4.2 png-apply: Write the working boilerplate to .png files

The `png-apply` command adds the current working boilerplate to one or more `.png` image files. It is usually used in this fashion:

```
$ licensing png-apply foo.png
png-apply: foo.png -> Boilerplate applied.
```

This example applies the current working boilerplate to `foo.png`.

The commenting style is automatically removed from the current working boilerplate before it is added to a `.png` file. This means that an *uncommented* form of the current working boilerplate is copied into the comment section of the PNG file.

This command will fail to operate when a copyright holder or a license have not been specified in the current working boilerplate. It will also fail if the given file is not recognized as a valid PNG file.

By default a backup `.bak` file is created; to prevent this behaviour use the `--no-backup` option.

The `--quiet` option prevents the ‘Boilerplate applied.’ messages from being displayed.

In the PNG file-format, there are named text sections. This command creates a named text section called "Comment" if it does not exist. Otherwise `png-apply` overwrites an existing one.

4.3 prepend: Add arbitrary text to the start of a file

When the current working boilerplate that `licensing` produces is insufficient in some way, the `prepend` command provides more control. For example, some free software developers prefer to change the GNU GPL license notice to replace ‘`This software`’ with the name of their program. Others may have different tastes in commenting styles or indentation. The purpose of this command is to allow that kind of extra control.

The `prepend` command is similar to the `apply` command but instead of adding the current working boilerplate to the beginning of source code files, an arbitrary file can be added. A file can be prepended in two ways. The first way looks like this:

```
$ licensing prepend myfile.txt foo.c --no-backup
```

In the second way, the same input file can be added via the standard input.

```
$ cat myfile.txt | licensing prepend foo.c --no-backup
```

These two commands are equivalent. Both add `myfile.txt` to the beginning of `foo.c`, while not retaining the original `foo.c` file in a backup file.

When the destination file happens to be a script, the input file will appear after the ‘#!’ line.

To place the input file after any previously existing boilerplate in the source file, use the `--after` option. Specify a commenting-style with one of the common options (see Chapter 9 [Common Commenting-style options], page 24, or the previously existing boilerplate in the destination file is automatically detected.

4.4 notice: Create and write a copyright and license notice to a file.

The `notice` command is a simple way to add a boilerplate to a set of source code files.

The copyright option `-c`, license option `-l` and commenting style option `-s` are *required options* in the `notice` command. When `-s` is specified more than once only the final option is used. Options `-c` and `-l` can be specified more than once to add more than one copyright holder, or more than one license notice.

The `notice` command is most often used like so:

```
$ licensing notice -c 'Yoyodyne,\ Inc.\ 2001' -l gpl -s c -n *. [ch]
apply: foo.c -> Boilerplate applied.
apply: bar.c -> Boilerplate applied.
apply: qux.c -> Boilerplate applied.
apply: foo.h -> Boilerplate applied.
```

This example creates a boilerplate and adds it to the beginning of all `.c` and `.h` files, while not retaining any backup files. Everything in the argument to `-c` is passed to the `copyright` command to operate on. The boilerplate includes the GNU GPL notice and a copyright line for Yoyodyne, Inc. The `-s` option is being used to set the commenting style to the C-style. The `-n` turns off backups. To see what licenses and styles are available, type ‘`licensing notice --help`’.

The backslashes in the `-c` command ensure that the argument is parsed as a single argument.

Often greater precision is required than the `notice` command provides. This command is included to offer a simpler way to add boilerplate in small cases.

When the copyright option `-c` is given an argument that results in an error in the `copyright` command (for example a malformed year specification), notices are not applied to any source code files.

5 Working with comments

These commands deal with comments.

5.1 `comment`: Create a comment block in a given style

The `comment` command makes a comment in a given style. If no style is given, the comment is created in the C-style. For example:

```
$ echo "foo" | licensing comment
/* foo */
```

The `comment` command can also operate on files specified as arguments, but it does not modify those files. The result is always displayed on the standard output. For example:

```
$ echo "foo" > foo.txt
$ licensing comment foo.txt
/* foo */
```

To specify a particular commenting style, use one of the common comment-style options (see Chapter 9 [Common Commenting-style options], page 24). For example:

```
$ echo "foo" | licensing comment --c++-style
// foo
```

This example makes a comment in the C++ style.

The `preview` command uses `comment` to help generate the current working boilerplate.

5.1.1 Dealing with indentation

The `comment` command tries to retain existing indentation by pushing the whole text to the right. When a block-style comment is being created the text is pushed three columns to the right. In the case of whole-line comments, the number of columns pushed is the size of the comment delimiter plus one.

```
$ cat foo.txt
    foo
    bar
$ licensing comment foo.txt
/*      foo
       bar */
```

This example shows that 3 columns are added to the beginning of every line. The `foo.txt` file has four spaces of indentation, and the result moves the ‘foo’ and ‘bar’ to the seventh column.

When a block-style comment is being created, and the final line of the comment is over 78 characters, the closing delimiter is placed on the following line by itself.

6 License commands

These commands display licenses and license notices. Licenses are the full terms and conditions, while notices are simply disclaimers about which license is being used. Some licenses are small enough to not require a license notice.

There could be a slight delay in the operation of these commands because licenses and license notices are fetched on-demand from the www.gnu.org or www.fsf.org websites.

Each of these license commands provides one or more license keywords to the `choose` command. To see the license keywords that a command provides, run ‘`licensing COMMAND --list-license-notices`’. For example:

```
$ licensing gpl --list-license-notices
gplv3+,full-license
gplv2+,full-license
gplv1+,full-license
gplv3+
gplv3,no-later
gplv2+
gplv2,no-later
gplv1+
gplv1,no-later
gplv1,no-later,mass
gplv1+,mass
gplv2,no-later,temple
gplv2+,temple
gplv2,no-later,franklin
gplv2+,franklin
gplv3,no-later,franklin
gplv3+,franklin
```

This example shows the licensing keywords that the `gpl` command provides to the `choose` command. When a license is not recommended, the `--force` option must be used. For example:

```
$ licensing choose gplv3,no-later
choose: ‘gplv3,no-later’ is unrecommended! Use --force, or maybe try ‘gplv3+’
```

Although the GNU General Public License version 3 is a fine license, it is not recommended to remove the *Or any later version* clause because future improvements to the GPL cannot be automatically granted.

The `,franklin`, and `,temple` texts refer to mailing addresses for the Free Software Foundation. Nowadays most projects forego the mailing address for a hypertext link.

6.1 `gpl`: Show the GNU General Public License notice

The `gpl` command displays the license notice for the latest version of the GNU General Public License. Options `--v1`, `--v2`, and `--v3` control which version of the license is shown.

To display the full license text instead of just the license notice, use the `--full` option.

The `--jerkwad` option removes the *Or any later version* clause from the license notice.

This command downloads the license from www.gnu.org/licenses/gpl.txt.

The address options `--address-link`, `--address-franklin`, `--address-temple`, and `--address-mass` are used to modify the final paragraph of the license notice to point to the website, the current address, or an old address of the Free Software Foundation. The old addresses are added for license notice matching on old source files which may not have updated their addresses.

6.2 `lgpl`: Show the GNU Lesser General Public License notice

The `lgpl` command displays the license notice for the latest version of the GNU Lesser General Public License. Options `--v2`, `--v2.1`, and `--v3` control which version of the license is shown.

To display the full license text instead of just the license notice, use the `--full` option.

The `--jerkwad` option removes the *Or any later version* clause from the license notice.

This command downloads the license from www.gnu.org/licenses/lgpl.txt, except for version 3 of the license notice. A local copy of the version 3 license notice is used.

6.3 `agpl`: Show the GNU Affero General Public License notice

The `agpl` command displays the license notice for the GNU Affero General Public License.

To display the full license text instead of just the license notice, use the `--full` option.

The `--jerkwad` option removes the *Or any later version* clause from the license notice.

This command downloads the license from www.gnu.org/licenses/agpl.txt.

6.4 `fdl`: Show the GNU Free Documentation License notice

The `fdl` command displays the license notice for the latest version of the GNU Free Documentation License. Options `--v1.1`, `--v1.2`, and `--v1.3` control which version of the license is displayed.

To display the full license text instead of just the license notice, use the `--full` option.

The `--jerkwad` option removes the *Or any later version* clause from the license notice.

This command downloads the license from www.gnu.org/licenses/fdl.txt.

6.5 `all-permissive`: Show the GNU All-Permissive License

The `all-permissive` command displays the GNU All-Permissive License.

This license is short enough to also serve as a license notice.

`all-permissive` downloads the license from www.gnu.org/prep/maintain/html_node/License-Notices-for-Other-Files.html.

6.6 fsf-permissive: Show the FSF Permissive License

The `fsf-permissive` command displays the FSF Permissive License. It is used primarily in the `autoconf` project.

There is also a short version of this license that can be seen with the `--brief` option.

Both form of this license are short enough to serve as a license notice.

There is no canonical place for this license, so `licenseutils` provides it internally.

6.7 bsd: Show the Berkeley Software Distribution license

The `bsd` command displays the Berkeley Software Distribution License. The options `--2-clause` and `--3-clause` control which variant of the license is displayed.

This license is short enough to also serve as a license notice.

This command downloads the license from `directory.fsf.org/wiki/License:BSD_3Clause`.

By default the three clause BSD license is shown. This license is easily confused with the OpenBSD license. See Section 6.10 [isc invocation], page 19, for more information.

6.8 apache: Show the Apache License notice

The `apache` command displays the license notice for the latest version of the Apache License.

To display the full license text instead of just the license notice, use the `--full` option.

This command downloads the Apache License from `directory.fsf.org/wiki/License:Apache2.0`.

6.9 mit: Show the MIT license

The `mit` command displays the Massachusetts Institute of Technology License. It is also known as the X11 License.

This license is short enough to also serve as a license notice.

`mit` downloads the license from `directory.fsf.org/wiki/License:X11`.

6.10 isc: Show the ISC license

The `isc` command displays the Internet Systems Consortium License. It is also known as the OpenBSD License.

This license is short enough to also serve as a license notice.

`isc` downloads the license from `directory.fsf.org/wiki/License:ISC`.

6.11 artistic: Show the Artistic license notice

The `artistic` command displays the notice for the Artistic License. It is the license most popularly associated with the `perl` project.

This license is not short enough to serve as a license notice, and there is no standardized notice, so one is supplied from within the `licenseutils` project.

There are two versions of the license: 1.0 and 2.0, and only the 2.0 is compatible with the GPL. The full-text of the license can be seen using the `--full` option. Version 1.0 can be seen with the `-1` option.

`artistic` downloads the full-text of the license from `directory.fsf.org/wiki/License:ArtisticLicense2.0`.

6.12 `ep1`: Show the Eclipse Public License notice

The `ep1` command displays the notice for the Eclipse Public License. It is the license most popularly associated with the eclipse project.

The license notice for the `ep1` is provided from within the `licenseutils` project, and is modeled after the notices found in `eclipse`.

There are two versions of the license: 1.0 and 2.0, and they are both incompatible with the GPL. The full-text of the license can be seen using the `--full` option. Version 1.0 can be seen with the `-1` option.

`ep1` downloads the full-text of the license from `www.eclipse.org/org/documents/epl-2.0/EPL-2.0.txt`.

6.13 `mp1`: Show the Mozilla Public license notice

The `mp1` command displays the notice for the Mozilla Public License. It is the license most popularly associated with the Firefox project.

There are two versions of the license: 1.1 and 2.0, and version 1.1 is incompatible with the GPL. The full-text of the license can be seen using the `--full` option. Version 1.1 can be seen with the `-1` option.

`mp1` downloads the full-text of the license from `www.mozilla.org/media/MPL/2.0/index.txt`.

6.14 `zlib`: Show the ZLib license

The `zlib` command displays the ZLib license. It is most popularly associated with the `zlib` compression library.

This license is short enough to also serve as a license notice.

`isc` downloads the license from `directory.fsf.org/wiki/License:Zlib`.

7 Informational commands

These commands just display text of one sort or another.

7.1 preview: Show the current working boilerplate

The current working boilerplate is sort of a notional concept. It doesn't exist as a file anywhere until the `preview` command generates it. The `preview` command simply displays the current working boilerplate on the standard output.

It will fail to display when license has not been selected (with the `choose` command, see Section 3.2 [choose invocation], page 8), or a copyright holder hasn't been specified yet (with the `copyright` command, see Section 3.3 [copyright invocation], page 9.)

Here is an example of the `preview` command:

```
$ licensing choose all-permissive
Selected.
$ licensing copyright Yoyodyne, Inc. 2001 --quiet
$ licensing preview
Copyright (C) 2001 Yoyodyne, Inc.
```

```
Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved. This file is offered as-is,
without any warranty.
```

In this example we see the `preview` command displaying a simple all-permissive boilerplate.

It is natural to run this command often after making changes to the current working boilerplate. To see which commands can change the current working boilerplate see Chapter 3 [Creating boilerplate], page 8. Some users may wish to pipe the output of `preview` through `less` when it is large; for example: `'licensing preview | less'`.

The `--no-commenting-style` option is used to display the current working boilerplate in an uncommented form, even when a `commenting-style` has already been specified.

The `apply` and `png-apply` commands rely on the `preview` command to generate the current working boilerplate before writing it to files.

7.2 welcome: Show the welcome message

The `welcome` command is displayed when the `lu-sh` shell is started in interactive-mode. It looks like this:

```
licenseutils 0.0.8pre
Copyright (C) 2013, 2017 Ben Asselstine
This is free software with ABSOLUTELY NO WARRANTY.
For warranty details type 'warranty'.
For a list of commands type 'help'.
```

To start the `lu-sh` shell without the welcome message, run `'licensing --quiet'`.

7.3 warranty: Show the warranty message

The `warranty` command displays the version and license of `Licenseutils`, which includes a disclaimer about the warranty of the software. It is included for completeness, and also the because the GPL suggests it be present.

7.4 help: Get help on commands

The `help` command displays a list of all commands, and also can display the `--help` of individual commands.

To see a list of all of the commands that `licensing` accepts, type `'licensing help'`. The list of commands will be separated into license commands, and the rest.

To get more help on a specific command, type `'licensing help COMMAND'`. For example:

```
$ licensing help all-permissive
Usage: all-permissive [OPTION...]
Show the GNU All-Permissive License.
```

```
-?, --help           give this help list
--usage              give a short usage message
```

This example displays the help for the `all-permissive` command. It is equivalent to running `'licensing all-permissive --help'`.

8 Other commands

These commands just display text of one sort or another.

8.1 `forget`: Clear the downloaded-files cache

When a license command runs for the first time, the license is downloaded from the internet from its canonical source. These web pages end up in a cache, and the `forget` command erases the cache.

8.2 `detect`: Determine the license notice in a file

Display the license notice similarities. This command checks to see how similar the uncommented boilerplate of a file is to all of the licenses that `licenseutils` knows about. With the `-a` option, this command generates a list of percentages, one line for every license.

Each generated percentage means that the particular license notice is a given percentage similar to the uncommented boilerplate. When the percentage is 100 percent, it means `licenseutils` is certain of a match. When the percentage is less than 100%, it means that `licenseutils` is not certain of a match, but the string is similar to an extent. When two or more licenses are shown as 100% it means that the file has two or more license notices in it.

The `detect` command also checks for the full-text of licenses. And optionally, with the `--mention` option it will check for references to licenses. For example: *This program is licensed under the GPL.*

When a `.png` file is encountered, it is automatically checked by the `detect` command.

9 Common command-line options for commenting-style detection

These options appear in the `boilerplate`, `cbb`, `comment`, and `apply` commands.

`--c-style`

For `boilerplate` and `cbb`, this option detects comments at the start of the file in blocks like: `/* this */`, or whole-line comments like `// this`.

`--c++-style`

Same as `--c-style`.

`--javascript-style`

For `boilerplate` and `cbb`, this option detects comments at the start of the file in blocks like: `/* this */`, or whole-line comments like `// this`.

`--shell-style`

For `boilerplate`, and `cbb`, this option detects comments at the start of the file in whole-line comments that look like `# this`.

`--scheme-style`

For `boilerplate`, and `cbb`, this option detects comments at the start of the file in whole-line comments that look like `; this`.

`--texinfo-style`

For `boilerplate`, and `cbb`, this option detects comments at the start of the file in whole-line comments that look like `@c this` or `@comment this`.

`--latex-style`

Like `texinfo` style.

`--m4-style`

For `boilerplate`, and `cbb`, this option detects comments at the start of the file in whole-line comments that look like `# this`, or `dnl this`.

`--haskell-style`

For `boilerplate`, and `cbb`, this option detects comments at the start of the file in blocks like `{- this -}`, or whole-line comments like `-- this`.

`--groff-style`

Groff is the GNU program for making man pages. It is also called troff. For `boilerplate`, and `cbb`, this option detects comments at the start of the file in whole-line comments that look like: `\" this`, `.\" this`, `\" this`, or `.\" this`.

`--gettext-style`

For `boilerplate`, and `cbb`, this option detects comments at the start of the file in whole-line comments that look like `# this`. The space after `#` is required, as `#,` and `#:` are valid gettext symbols and not comments.

`--fortran-style`

For `boilerplate`, and `cbb`, this option detects comments at the start of the file in whole-line comments that look like `C this`, or `! this`.

--pascal-style

For `boilerplate`, and `cbb`, this option detects comments at the start of the file in blocks like `(* this *)`, or `{* this *}`, or whole-line comments like `// this`.

For the `apply` command, these options put the current working boilerplate after the existing boilerplate in the given style.

For the `comment` command, this option creates a comment in the given style.

There is a `--style` option that requires the name of a commenting style as an argument. This option does not appear in the `--help` of commands, but it still works.

For the sake of brevity, only a subset of the commenting styles are shown in the `--help` of commands. The `--help-all-styles` option will display all of the supported commenting style options.

Appendix A GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.
<http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

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 non-commercially. Secondly, 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.

The “publisher” means any person or entity that distributes copies of the Document to the public.

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 under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

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

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

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

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. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

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 1.3  
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.

Index

A

add a commenting style to the current working boilerplate	8
add a copyright holder to the current working boilerplate	9
add a file to the start of another file	14
add a license to the current working boilerplate	8
add a project-identifying line to the current working boilerplate	11
add a some extra text to the current working boilerplate	12
add an initial line to the current working boilerplate	11
add boilerplate to a .png file	14
add boilerplate to a source file	13
agpl	18
AGPL	18
all-permissive	18
apache	19
Apache license notice, showing the	19
Apache License, showing the	19
apply	13
artistic	19
Artistic License, showing the	19

B

Berkeley Software Distribution license, showing the	19
boilerplate	5
boilerplate, creation of	8
boilerplate, scanning of	5
boilerplate, writing of	13
bsd	19
BSD	19

C

cbb	6
choose	8
clear the downloaded-files cache	23
clearing the cache	23
comment	16
comments, creating	16
comments, removing delimiters	16
configuration files	1
copyright	9
count boilerplate blocks in source files	6
count lines of boilerplate in source files	6
create a comment block	16
creating boilerplate	8
current working boilerplate	1

current working boilerplate, adding a copyright notice	9
current working boilerplate, adding a license notice	8
current working boilerplate, adding an initial line	11
current working boilerplate, adding extra text ..	12
current working boilerplate, adding project identifier	11
current working boilerplate, adding to a png file	14
current working boilerplate, adding to a source file	13
current working boilerplate, changing the commenting style	8
current working boilerplate, creating a new	8
current working boilerplate, resetting the	8
current working boilerplate, showing the	21

D

detect	23
detecting licenses	23
detecting the license notice in a file	23

E

Eclipse Public License, showing the	20
ep1	20
erasing the current working boilerplate	8
extra	12

F

fd1	18
FDL	18
forget	23
fsf-permissive	19
FSF Permissive License, showing the	19

G

getting help on commands	22
GNU AGPL notice, showing the	18
GNU AGPL, showing the	18
GNU All-Permissive License, showing the	18
GNU FDL notice, showing the	18
GNU FDL, showing the	18
GNU GPL notice, showing the	17
GNU GPL, showing the	17
GNU LGPL notice, showing the	18
GNU LGPL, showing the	18
gp1	17
GPL	17

H

help 22

I

informational commands 21
 isc 19
 ISC 19
 ISC license, showing the 19

L

lgpl 18
 LGPL 18
 license commands 17
 license utilities 1
 license, displaying 17
 license, showing 17
 licensing 1

M

make a copyright notice 9
 mit 19
 MIT 19
 MIT license, showing the 19
 Mozilla Public License, showing the 20
 mpl 20

N

new-boilerplate 8
 notice 15

O

other commands 23

P

png-apply 14
 png-boilerplate 7
 prepend 14
 preview 21
 project 11

R

remove a copyright holder from the current
 working boilerplate 9
 remove boilerplate from .png files 7
 remove boilerplate from source files 5

S

Scanning for boilerplate 5
 select a commenting style 8
 select a license 8
 show boilerplate in .png files 7
 show boilerplate in source files 5
 show the current working boilerplate 21
 start a new boilerplate 8

T

top 11

W

warranty 22
 welcome 21
 writing boilerplate 13

Z

zlib 20
 ZLib license, showing the 20