

# Grayupload

---

version 1.1, 7 January 2024

**Sergey Poznyakoff.**

---

Copyright © 2023–2024 Sergey Poznyakoff

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, and no Back-Cover texts. A copy of the license is included in the section entitled “GNU Free Documentation License.”

## Table of Contents

<b>1</b>	<b>Overview .....</b>	<b>1</b>
<b>2</b>	<b>Quick Start .....</b>	<b>1</b>
<b>3</b>	<b>Introduction .....</b>	<b>2</b>
<b>4</b>	<b>Definitions .....</b>	<b>2</b>
<b>5</b>	<b>Destinations .....</b>	<b>3</b>
<b>6</b>	<b>Tarballs .....</b>	<b>4</b>
<b>7</b>	<b>Creating symlinks .....</b>	<b>5</b>
<b>8</b>	<b>Standalone Options .....</b>	<b>6</b>
<b>9</b>	<b>Debugging .....</b>	<b>7</b>
<b>10</b>	<b>Configuration .....</b>	<b>7</b>
<b>11</b>	<b>Release Types .....</b>	<b>9</b>
11.1	Version semantics .....	9
<b>12</b>	<b>Options .....</b>	<b>10</b>
12.1	Destination options .....	10
12.2	General options .....	10
12.3	File selection options .....	11
12.4	Symlink creation options .....	11
12.5	Informative options .....	12
<b>13</b>	<b>Bug Reports .....</b>	<b>12</b>
	<b>Appendix A GNU Free Documentation License ..</b>	<b>12</b>
	<b>Index .....</b>	<b>20</b>

## 1 Overview

The final step in the development cycle of a software package is uploading it to the server which will make it available to its users. For GNU software, this is normally done using the Section “Automated FTP Uploads” in *Information for Maintainers of GNU Software*, a protocol allowing the maintainer to distribute his work without intervention of system administrators. This protocol is used, for example, by ftp.gnu.org, alpha.gnu.org, and download.gnu.org.ua. Traditionally, the `gnupload` script was used to upload release tarballs to the destination server (see Section “gnupload” in *GNU Gnulib*). It is a simple script that relies on binaries normally found on each GNU/Linux system to do its job.

`Grayupload` is just another tool for this task. It was created as a result of experience gained during two decades of using `gnupload`. `Grayupload` was created with the following objectives in mind:

1. Minimize dependencies on external binaries.  
The only dependency left is `gpg`, which is used to sign the releases.
2. Provide a consistent command line interface.
3. Provide extensive debugging capabilities.
4. Ensure program extensibility.
5. Ensure smooth transition from `gnupload`.

## 2 Quick Start

The interface of `grayupload` is deliberately designed to be as close as possible to that of `gnupload`, so that in most cases it can be used as a drop-in replacement for the latter. Suppose a programmer maintains a package called ‘foo’ and distributes its tarballs from ‘<https://download.gnu.org.ua/release/foo>’. This means that the project is already registered, and the programmer has already uploaded his public GPG key to its account on ‘[puszcza.gnu.org.ua](https://puszcza.gnu.org.ua)’.

Now, to upload a new release (say `foo-1.0.tar.gz`), the programmer would run:

```
grayupload --to download.gnu.org.ua:ftp/foo foo-1.0.tar.gz
```

To create the symlink `foo-latest.tar.gz`, pointing to the newly uploaded file, he would run instead:

```
grayupload --to download.gnu.org.ua:ftp/foo --latest foo-1.0.tar.gz
```

Supposing the file `foo-0.9.tar.gz` is already available from the distribution server, one can create a symbolic link `foo-prior.tar.gz` pointing to it, by running

```
grayupload --to download.gnu.org.ua:ftp/foo \  
--symlink foo-0.9.tar.gz foo-prior.tar.gz
```

To remove an obsolete or broken tarball `foo-0.1.tar.gz` from the distribution server, one would run

```
grayupload --to download.gnu.org.ua:ftp/foo --delete foo-0.1.tar.gz
```

Similarly, if the distribution tarballs were distributed on ‘[ftp.gnu.org](https://ftp.gnu.org)’, the `--to` option in the examples above would have been rewritten as:

```
--to ftp.gnu.org:foo
```

These and other `grayupload` options are discussed in detail in the rest of this document.

## 3 Introduction

**Grayupload** takes all necessary information from arguments supplied in the command line. In general, these fall into two categories: options, and their arguments.

*Option* is a letter that begins with a single dash character, or a word that begins with a two dashes. The former form is called *short option* and the latter *long option*. If two or more short options are used in a row, they can be *clustered*, i.e. placed one after another after a single dash character. E.g. `-nvr` is equivalent to `-n -v -r`.

Long option names can be supplied in full, or in abbreviated form. **Grayupload** accepts any non-ambiguous abbreviation.

Some options require one or more *arguments*. If a long option takes single argument, this argument can be given either as the next word after the option, or together with the option name, delimited from it with a single equals sign: `--user foo` or `--user=foo`.

For a short option that takes single argument, the argument can either be given as the next word after the option, or immediately after the option letter, so that `-t stable` and `-tstable` are equivalent.

If an option takes multiple arguments, all words that follow it and don't start with a dash are treated as its arguments. In other words, anything up to the next option or end of command line is collected as option arguments.

A double-dash alone signals end of options. Anything after it is understood as non-option arguments.

Normally, each invocation supplies at least one *destination* option and a tarball. For example:

```
grayupload --to download.gnu.org.ua:ftp/bar bar-0.1.tar.gz
```

**Grayupload** will form a *triplet* by creating a detached PGP signature of the tarball and a clearsigned *directive* file and upload the three files to the server supplied in the destination option (`--to`, in this example).

If the release tarball already exists on the distribution server, it is an error to try to upload it again. However, if you are really sure you want to replace it with a newly created one, you can do so by supplying the `--replace` option. Depending on the upload server setup it will either remove the old tarball or move it away to the archive storage prior to the actual upload.

## 4 Definitions

In this document, the following definitions are used:

### *distribution server*

A server from which release tarballs are distributed. Usually, it uses HTTPS, HTTP, or anonymous FTP. Examples of distribution servers are: `ftp.gnu.org`, `alpha.gnu.org`, or `download.gnu.org.ua`.

### *distribution directory*

A directory on the *distribution server* from which *release tarballs* are available for download.

*upload server*

A server that accepts uploaded files, verifies and authorizes the upload, and makes *release tarballs* available from the *distribution server*. An example of upload servers is ‘`ftp-upload.gnu.org`’. An upload server may be combined with the distribution one, as is the case with ‘`download.gnu.org.ua`’.

*upload directory*

A directory on *upload server* where files can be uploaded.

*destination*

A URL specifying the *upload server*, *upload directory* and, *destination directory*.

*release tarball*

A tar archive to be made available from the *distribution server*.

*tarball signature*

A detached PGP signature of the *release tarball* as described in Section “FTP Upload Release File Triplet” in *Information for Maintainers of GNU Software*.

*directive file*

A clearsigned plaintext file describing the upload. This file contains instructions for the *upload server* on what to do with the uploaded files. See Section “FTP Upload Directive File” in *Information for Maintainers of GNU Software*, for a detailed description.

*triplet*

A set of three files as defined in Section “FTP Upload Release File Triplet” in *Information for Maintainers of GNU Software*: *tarball*, *detached tarball signature*, and clearsigned *directive file*. file.

## 5 Destinations

Upload *destination* defines, directly or indirectly, distribution server and distribution directory for the upload. At least one destination must be defined in each `grayupload` invocation. It is possible to supply several destination in one upload.

Two options are provided that supply destinations. The `--url` option defines the URL of the *upload server*, *upload directory* and, optionally *destination directory*, as well as the protocol used for upload. The URL argument has the following form:

```
scheme://[user@]host/[upload_dir] [;dest_dir]
```

Here, square brackets denote optional parts. The parts of a URL are:

*scheme* Defines the protocol used for upload. Following values are understood: ‘`ftp`’, for anonymous FTP upload, ‘`sftp`’, for upload via SFTP, and ‘`file`’, for copying triplet files to a local directory. In the latter case, the *user* and *host* parts must be omitted.

The ‘`scp`’ protocol is deliberately not implemented, due to associated security implications.

*user* Optional username for authorization. Used for ‘`ftp`’ and ‘`sftp`’. If not supplied, ‘`ftp`’ is used for ‘`ftp`’, and current login name for ‘`sftp`’.

*host*            Hostname of the upload server.

*upload\_dir*  
                  Upload directory.

*dest\_dir*        Destination directory.

For example, the following will form a triplet for uploading `tar-1.34.90.tar.gz` to the `/incoming/alpha` directory and upload it via anonymous FTP to server `'ftp-upload.gnu.org'`. The server will be instructed to make the uploaded tarball available from the `tar` subdirectory at the distribution server:

```
grayupload --url 'ftp://ftp-upload.gnu.org/incoming/alpha;tar' \
            tar-1.34.90.tar.gz
```

Using the `--url` option requires knowledge about upload server, which can be considered an “internal” aspect of the automatic upload system. Another option is available, which does not require this knowledge. The `--to` option takes as argument one of the *standard destinations*, which are well-known destinations built into `grayupload`. A standard destination specifies the *distribution server* and directory. As of version 1.1 of `grayupload`, the following standard destinations are understood:

`ftp.gnu.org:dir`

Upload for distribution from `'https://ftp.gnu.org/gnu/dir'`.

`alpha.gnu.org:dir`

Upload for distribution from `'https://alpha.gnu.org/gnu/dir'`.

`[user@]download.gnu.org.ua:ftp/dir`

Upload for distribution from `'https://download.gnu.org.ua/release/dir'`. If your remote and local user names differ, use the `'user@'` prefix.

`[user@]download.gnu.org.ua:alpha/dir`

Upload for distribution from `'https://download.gnu.org.ua/alpha/dir'`. If your remote and local user names differ, use the `'user@'` prefix.

`/dir[:dist_dir]`

Prepare the triplet for distribution from the directory `dist_dir` and copy it to the local directory `dir`. If `dist_dir` is omitted, `dir` is used instead.

This is a shortcut of the `file://` URL, useful mainly for debugging.

## 6 Tarballs

Any word in the command line that is not an argument to a preceding option specifies a release tarball for upload. Normally, tarballs are located in the current working directory. However, it is not required: they can reside anywhere on the local filesystem. The directory part is stripped off the tarball name before creating the `filename:` stanza in the directive file. Thus, the created directive will instruct upload server to place the file to the top-level directory of the project. For example, if you run

```
grayupload --to alpha.gnu.org:foo build/foo-0.1.tar.gz
```

then, as a result of the upload, the file `build/foo-0.1.tar.gz` will appear in `'https://alpha.gnu.org/gnu/foo/'`. Most servers allow uploaders to create any directory

hierarchy your their package directory. If you do wish to upload your tarballs to a subdirectory, give its name in the destination specification, for example:

```
grayupload --to alpha.gnu.org:foo/temp build/foo-0.1.tar.gz
```

The subdirectory will be created if it doesn't exist.

Notice, that `grayupload` creates signature and directive files in the directory where the tarball is located. This means, obviously, that this directory must be writable.

Created signature files remain on disk after termination of the program. You may need these for other tasks related to the release. In contrast, directive files are always removed. If you wish to inspect their content, use two `-v` options.

## 7 Creating symlinks

It is a customary practice to keep in the distribution directory a *latest* symlink pointing to the most recent tarball available. You can maintain such symlink using the `--latest` option when uploading. For example:

```
grayupload --to ftp.gnu.org:foo --latest foo-1.0.tar.gz
```

This command will upload `foo-1.0.tar.gz` and make it available from `'https://ftp.gnu.org/gnu/foo'`. At the same time, it will create a link `'https://ftp.gnu.org/gnu/foo/foo-latest.tar.gz'`, pointing to `'https://ftp.gnu.org/gnu/foo/foo-1.0.tar.gz'`. If the `foo-latest.tar.gz` link already existed, it will have been replaced with the new one.

A generalized version of this option is `--transform-symlink`. It takes as its argument a `sed`-like replace expression of the form:

```
s/regexp/replace/[flags]
```

where *regexp* is a *regular expression*, *replace* is a replacement for each file name part that matches *regexp*. Both *regexp* and *replace* are described in detail in Section “The ‘s’ Command” in *GNU sed*.

Multiple `sed` expressions can be supplied in one option by separating them with a semicolon.

When processing the `--transform-symlink` option, the expression is applied to each release tarball name to produce a symlink name. Its result, if it differs from the original name, and is not empty, gives the name of the symbolic link. If multiple expressions are given, each subsequent expression is applied to the result of the previous one. The last expression applied produces the symlink name.

Any delimiter can be used in lieu of `'/'`, the only requirement being that it be used consistently throughout the expression. For example, the following two expressions are equivalent:

```
s/one/two/
s,one,two,
```

Changing delimiters is often useful when the *regex* contains slashes. For example, the `--latest` option described above is just a shortcut for:

```
--transform-symlink 's|- [0-9] [0-9\.] * \ ( - [0-9] [0-9] * \) \ {0,1} \ . | -latest . | '
```

Supported *flags* are:

`'g'` Apply the replacement to *all* matches to the *regexp*, not just the first.



- ‘i’            Use case-insensitive matching.
- ‘x’            *regexp* is an *extended regular expression* (see Section “Extended regular expressions” in *GNU sed*).
- ‘*number*’    Only replace the *numberth* match of the *regexp*.  
If this flag is used together with ‘g’, this means ignore all matches before the *numberth* one and replace all matches from the *numberth* on.

## 8 Standalone Options

Standalone options don’t require a release tarball. They can be used to delete existing tarballs or symlinks from the distribution server or to create symlinks to already released tarballs<sup>1</sup>.

To delete an already existing tarball, use the `--delete` option:

```
grayupload --to ftp.gnu.org:foo --delete foo-1.0.tar.gz
```

Notice, that in this case `foo-1.0.tar.gz` is not the name of a tarball to be uploaded. It is the name of a tarball that already exists on the distribution server, and that the user wants to remove.

Depending on the distribution server setup, the tarball can be removed or *archived*, i.e. moved away to some archive storage, from where it can be restored if the need be.

A similar option, `--rmsymlink`, deletes an existing symlink:

```
grayupload --to ftp.gnu.org:foo --rmsymlink foo-latest.tar.gz
```

Both `--delete` and `--rmsymlink` options take multiple arguments, and can thus be used to delete multiple files (or symlinks) in one go.

Third standalone option, `--symlink`, allows you to create symbolic links to already existing release tarballs. It takes any number of ‘*tarball linkname*’ pairs as its arguments. Here, *tarball* is the name of an existing tarball, and *linkname* is the name of a symlink to that tarball that you want to create.

Although not designed for such usage, standalone options can be mixed with the regular uploads. If you do wish to use the two at once, use the `--upload` option to introduce release tarball names, e.g.:

```
grayupload --to ftp.gnu.org:foo \  
  --rmsymlink foo-latest.tar.gz foo-stable.tar.gz \  
  --upload foo-1.0.tar.gz --latest
```

The above example will cause two directive files to be created and uploaded: one containing standalone directives, and another one for uploading `foo-1.0.tar.gz`.

Notice, that in this case the remote server gives no promise on the exact order in which the submitted directives will be applied.

---

<sup>1</sup> Technically speaking, these implement standalone directives, as described in Section “FTP Upload Standalone Directives” in *Information for Maintainers of GNU Software*.

## 9 Debugging

To get more insight into what's going on when you upload a tarball, use the `-v` (`--verbose`) option. This option is incremental, that is the more times you supply it, the more verbosity you get as a result. The single `-v` option causes the program to verbosely annotate each step it is taking. Two `-v` options instruct it to additionally print the content of the produced directive files. Three options will additionally print a detailed protocol transcript.

Another option useful for debugging is `--dry-run` (`-n`). It instructs `grayupload` to do everything in verbose mode (single `-v` is implied), but not actually upload any files.

## 10 Configuration

If the file `.grayupload` exists in the current working directory, the program reads its options from it.

The lines in configuration file are processed sequentially. Empty lines are ignored. The `#` character starts inline comment: the character itself and all characters that follow it up to the end of line are ignored.

A non-empty file must consist of a keyword and value separated by any amount of whitespace. Leading and trailing whitespace is removed. If the value includes whitespace or comment characters, it should be enclosed in double quotes. Within double-quotes a backslash character serves as an escape indicator: it is removed and the character that follows it is taken literally. This allows for embedding double quotes (`\"`) and backslashes (`\\`) in quoted strings.

Allowed keywords correspond to option names without leading dash:

`to` *dest* [Config]

Defines destination location. Argument must be one of standard destinations (see [Standard destinations], page 4). It can optionally be prefixed with `'type='` to indicate that it is to be used only for the given release type. See Chapter 11 [Release types], page 9, for a detailed discussion of this useful feature.

The `--to` and `--url` command line options override this setting.

`url` *dest* [Config]

Defines destination URL. See [upload url], page 3, for a discussion of *dest* syntax. The URL can optionally be prefixed with `'type='` to indicate that it is to be used only for the given release type. See Chapter 11 [Release types], page 9, for a detailed discussion of this useful feature.

`directory` *dirname* [Config]

Sets the default destination directory name, This name will be used for any subsequent `url` statement that doesn't specify destination directory explicitly.

`user` *id* [Config]

Sign tarballs and directives with the GPG key *id*.

**comment** *text* [Config]

Add comment *text* to the directive. By default, single comment is added specifying the name and version of the program that did the upload. Multiple **comment** statements are allowed. Eventual **--comment** command line options don't override, but rather append their arguments to the list of comments created by the **comment** statements.

If *text* contains whitespace, double quotes or backslash characters, it must be enclosed in a pair of double quotes (and embedded double quotes or comments escaped as described at the beginning of this chapter).

**verbose** *number* [Config]

Set verbosity level. Allowed values for *number* are 0 through 3.

**version\_semantics** *name* [Config]

*Version semantics* defines the algorithm used to determine release type from version number. It is used if release type is not specified explicitly in the command line.

Possible values for the *name* argument are **gnu** and **kernel**. If version semantics is set to **gnu**, version numbers consisting of major and minor parts imply 'stable' release. If the version number has patch-level part, and this part is numerically greater than or equal to 90, then 'alpha' release is assumed. Otherwise, the version indicates 'stable' release.

If *name* is **kernel**, then major part of the release number determines the release type. If it is even, then it is a 'stable' release. Otherwise, it is a 'alpha' release.

See Section 11.1 [Version semantics], page 9, for a detailed discussion of this feature.

**latest** *bool* [Config]

If *bool* is 'true', then, for each file uploaded to the server, create the *latest* symlink. See Chapter 7 [Creating symlinks], page 5, for details.

Allowed *bool* values are **true**, **on**, **yes**, **1**, for 'true' and **false**, **off**, **no**, and **0**, for 'false'.

**transform\_symlink** *s-exp* [Config]

For each uploaded file, create a symlink, with the name obtained by applying a **sed** expression *s-exp* to the file name. See Chapter 7 [Creating symlinks], page 5, for details.

**replace** *bool* [Config]

If a tarball already exists on the distribution server, replace it with the supplied one. See the description of the **latest** statement for the list of allowed values for *bool*.

Command line options override the corresponding configuration statements. For example, if your **.grayupload** file contains three **to** statements and you supply a **--to** option in the command line, these configuration statements are ignored and the destination supplied in the command line is used instead.

To supply alternative configuration file, use the **--config** *file* option. If given, *file* will be read, instead of **.grayupload**. Notice, that *file* must exist.

To ignore existing configuration file, use the **--no-config** option.

## 11 Release Types

Software releases can be categorized by their *release type*, such as ‘**stable**’, ‘**alpha**’, etc. Tarballs of different release types normally go to different destinations. In order to make the release process more convenient, **grayupload** provides a mechanism for selecting the destination URL by the release type.

This works as follows. Destination locations supplied with the **to** and **url** configuration statements are prefixed (*qualified*) with the release type, as in:

```
to stable=ftp.gnu.org:myproject
to alpha=alpha.gnu.org:myproject
```

There are no predefined release types, it is up to the user to define as many of them as needed. A valid release type name must begin with an alphabet letter or underscore, and contain letters, digits, underscores or dashes.

When calling **grayupload**, you can either supply the actual release type with the **--release-type** (**-t**) option in the command line, or let the program determine it from the version number automatically.

```
grayupload --release-type=stable myproject-1.0.tar.gz
```

When invoked this way, **grayupload** scans each destination and selects only those of them, which are qualified with that release type (‘**stable**’, in this case).

Both qualified and unqualified destinations can be listed together in the configuration.

The exact matching rules are:

1. Unqualified destinations always match.
2. Qualified destinations match only if the release type is determined (either using the **--release-type** option, or derived from the version number) and its value matches the destination qualifier.

### 11.1 Version semantics

Release version numbers can often be used to derive the intended release type. The algorithm used to do so depends on the *version semantics*. As of version 1.1, **grayupload** supports two distinct semantics: **gnu** and **kernel**. Both assume that the version number consists at least of two decimal numbers, *major* and *minor*, delimited by a single dot. The minor number can be followed by another dot and a decimal number, called *patch-level*.

The **gnu** version semantics is used by GNU projects. If the version number contains only major and minor parts, the release type is ‘**stable**’. If the patch-level part is present, its value is taken into account. If it is greater than or equal to 90, this is an ‘**alpha**’ release. Otherwise, this is a ‘**stable**’ release.

Another version semantics is **kernel**, called so because it is used in Linux kernel version numbering. When this semantics is selected, release type is determined by the version major number: even major numbers correspond to ‘**stable**’ releases, whereas odd ones correspond to ‘**alpha**’ releases.

To determine release types automatically, use the **version\_semantics** statement in your configuration file (see Chapter 10 [Configuration], page 7) to define the version semantics. For example, to use ‘**gnu**’ semantics:

```
version_semantics gnu
```

When this statement is present, `grayupload` scans the list of upload file names. It extracts the version number from each file name argument and tries to determine release type from it using the requested semantics. If all file names yield the same release type, it is used to determine the upload destination as described in Chapter 11 [Release types], page 9.

Notice the following important points. First of all, version semantics is applied only if the release type is not specified explicitly in the command line (using the `--release-type` option). Secondly, it applies only if at least one file to be uploaded is supplied in the command line. In particular, it won't be applied if only standalone options are present in the command line (see Chapter 8 [Standalone Options], page 6). And finally, if more than one upload file is given, version semantics is used only if it produces the same result when applied to each file name.

## 12 Options

This chapter summarizes the available command line options.

### 12.1 Destination options

`--to target:dir`

Use a *standard destination* specification. This makes the tarballs available from the distribution server *target*, directory *dir*.

See [Standard destinations], page 4, for a detailed description of the available standard destinations.

`--url url` Upload to the supplied *url*. See Chapter 5 [Destinations], page 3, for details.

`--directory dir`

This option sets the default destination directory name. This name will be used for any subsequent `--url` option that doesn't specify destination directory explicitly.

`--release-type type`

`-t type` Select only destinations marked with this release type. See Chapter 11 [Release types], page 9, for a detailed discussion of this option.

### 12.2 General options

`--config=file`

Read configuration from file *file*, instead of the default `.grayupload`. The file must exist and be readable.

See Chapter 10 [Configuration], page 7, for a detailed discussion.

`--no-config`

Ignore existing `.grayupload` file.

See Chapter 10 [Configuration], page 7, for a detailed discussion.

**-n**  
**--dry-run** Enables *dry-run* mode: don't upload any files, just print what would have been done. See Chapter 9 [Debugging], page 7.

**-v**  
**--verbose** Increase debugging verbosity level. The option is incremental: the more times you repeat it the more verbosity you get. See Chapter 9 [Debugging], page 7, for a detailed discussion.

**-u**  
**--user *id*** Sign tarballs and directives with the GPG key *id*.

**--comment *text*** Add comment *text* to the directive. By default, single comment is added specifying the name and version of the program that did the upload. Multiple **--comment** options are allowed.

### 12.3 File selection options

The following options select files or symlinks to be created or removed. See Chapter 8 [Standalone Options], page 6, for a detailed discussion.

**--delete *file...***  
Delete these files. One or more arguments are allowed.

**--rmsymlink *file...***  
Delete these symbolic links. One or more arguments are allowed.

**--symlink *file symlink ...***  
Create a symbolic link *symlink* pointing to the release tarball *file*. Any number of *file symlink* pairs is accepted.

**--upload *file...***  
Upload files. Use this option after any of the above options to mark end of arguments to the previous option, and start of file names for upload.

**--replace**  
If a tarball already exists on the distribution server, replace it with the supplied one. See Chapter 3 [Introduction], page 2.

### 12.4 Symlink creation options

These are discussed in detail in Chapter 7 [Creating symlinks], page 5.

**--latest** For each file uploaded to the server, create the *latest* symlink. The name of the symlink is obtained by replacing version number in the file name with the word '-latest'. For example, link name for the archive `foo-1.0.tar.gz` will be `foo-latest.tar.gz`.

**--transform-symlink *s-exp***  
For each uploaded file, create a symlink, with the name obtained by applying a `sed` expression *s-exp* to the file name.

`--symlink-regex[=s-exp]`

This option is provided for compatibility with the `gnupload` script. It is equivalent to `--transform-symlink s-exp`, if *s-exp* is supplied, and to `--latest` otherwise. Notice, that the argument, if present, must be delimited from the option name by a single equals sign with no additional whitespace on either side of it.

## 12.5 Informative options

These options cause the program to print the requested information and exit successfully.

`-?`

`--help`     Display a short usage summary.

`-V`

`--version`

Display program version and distribution license.

## 13 Bug Reports

If you think you found a bug in `grayupload` or in its documentation, please send a mail to `gray@gnu.org` (Sergey Poznyakoff) or use the bug tracker at <https://puszcza.gnu.org.ua/bugs/?group=grayupload> (requires authorization).

## Appendix A GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.  
<https://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 <https://www.gnu.org/licenses/>.

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

—		<b>D</b>	
--delete .....	6	delete a distributed tarball .....	6
--dry-run .....	7	destination .....	3
--latest .....	5	directive file .....	3
--replace .....	2	directory .....	7
--rmsymlink .....	6	distribution directory .....	2
--symlink .....	6	distribution server .....	2
--to .....	4	download.gnu.org.ua .....	4
--transform-symlink .....	5		
--upload .....	6	<b>F</b>	
--url .....	3	ftp.gnu.org .....	4
--verbose .....	7		
		<b>G</b>	
•		gnu, version semantics .....	9
.grayupload .....	7		
<b>A</b>		<b>K</b>	
alpha.gnu.org .....	4	kernel, version semantics .....	9
<b>C</b>		<b>L</b>	
comment .....	8	latest .....	8
creating symlinks .....	6		

**R**

release tarball ..... 3  
replace ..... 8

**S**

sed ..... 5  
standard destinations ..... 4  
symlink, latest ..... 5  
symlinks, creating ..... 6

**T**

tarball signature ..... 3  
to ..... 7  
transform\_symlink ..... 8  
triplet ..... 3

**U**

upload directory ..... 3  
upload server ..... 2  
upload URL ..... 3  
url ..... 7  
URL ..... 3  
user ..... 7

**V**

verbose ..... 8  
version\_semantics ..... 8  
version\_semantics, described ..... 9