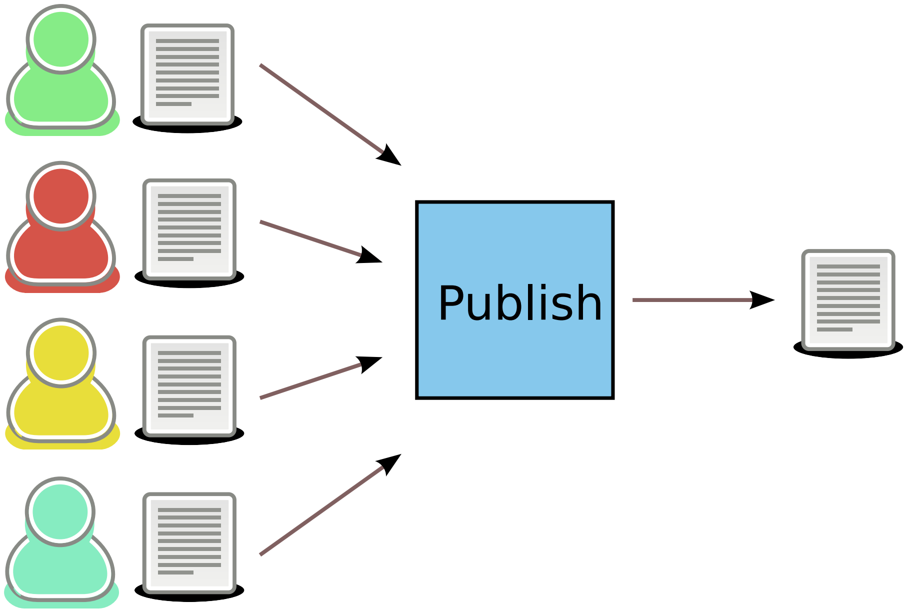
Publish User Manual

April 15, 2013



Anna Logg

www.loggsystems.se

Contents

Chapter 1

Introduction

Researchers and research institutions are faced with an increasing demand for demonstrating their productivity, for example when applying for funding. Demonstrating productivity means preparing publication records, for inclusion in a CV, on a web page, or in an annual report. Preparing such publication records is time-consuming and error-prone, in particular when a large number of publications must be presented in a uniform manner.

The bibliographic reference system **Publish** solves this. Publication records (in for example BibTeX format) from a large number of researchers or departments may be imported into a common database, validated against a list of known venues (journal and conference names), checked for duplicate entries and common typos. Publication records may then be categorized and generated, in BibTeX, LaTeX, or PDF format.

The database is maintained as a simple text file which may be easily edited using any text editor.

This manual describes the command-line interface of **Publish**. In addition, a programmer's interface is provided in the form of a Python module. For documentation of the Python interface, refer to the Python help system[1].

[1] `help(publish)`

Chapter 2

Overview

Publish handles the following three basic tasks: importing, validating, and exporting papers. We discuss these briefly below and return to discuss each of them in the following chapters.

2.1 Importing Papers

When importing a list of papers, they end up in a publication database stored as a simple text file. The file is stored in the current working directory and is named `papers.pub`. When new data is imported into the database, both the database and the imported data are validated. The two lists of papers are subsequently merged (with check for duplicates) and stored back to the database, as illustrated in Figure 2.1.

For more details, see Chapter 4.

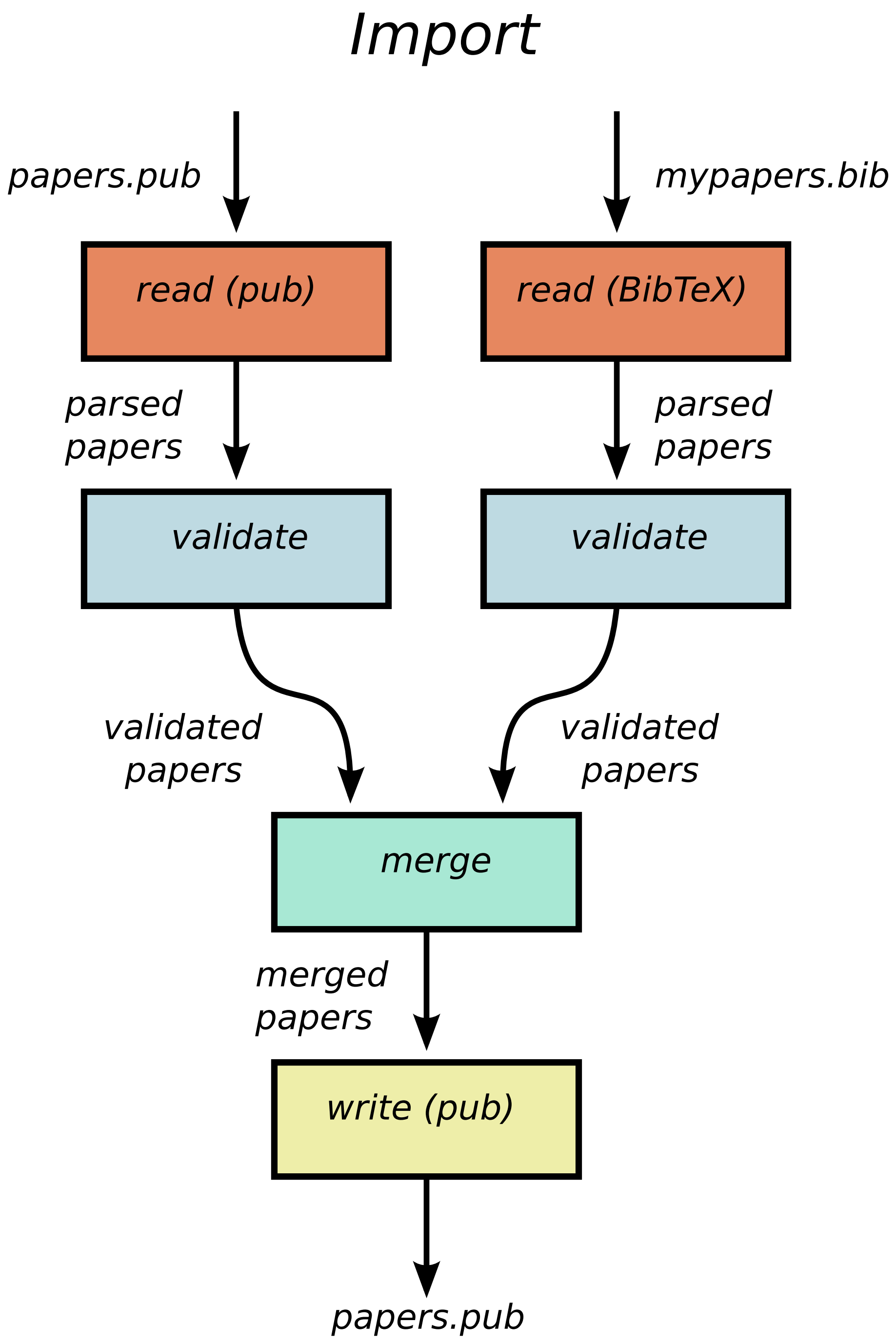


Figure 2.1: Importing papers.

2.2 Validating Papers

After data has been imported into the database, it is possible to validate the database for errors. Validation happens automatically when data is imported, so this feature is mostly relevant when the database file is edited directly. A schematic overview of the validation process is given in Figure 2.2.

For more details, see Chapter 5.

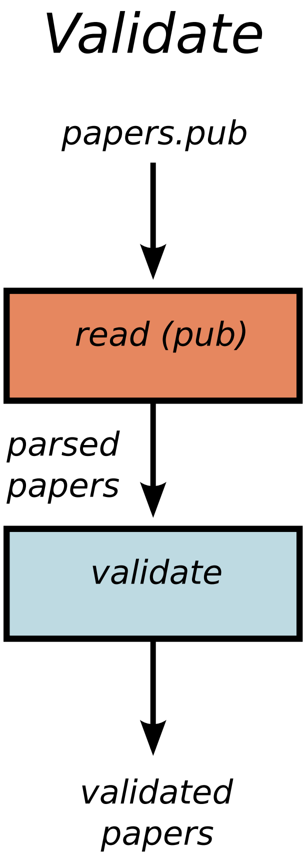


Figure 2.2: Validating papers.

2.3 Exporting Papers

Publication records may be exported from the database in one of the following formats: `pub`, BibTeX, LaTeX, or PDF. The user may also specify a filter to export a subset of the papers in the database, for example all papers in a specific category or from a specific year. During export, all papers are first read from the database, then validated, then filtered, and then formatted, as illustrated in Figure 2.3.

For more details, see Chapter 6.

2.4 The `pub` Format

The system uses its own format for storage of the papers. The format is designed to allow simple editing[1] and looks as follows:

```
* category
** title
   attribute: value
   attribute: value
   ...
** title
   attribute: value
   attribute: value
   ...
* category
   ...
```

See Appendix C for more information about the `pub` format.

[1] The format is based on the Emacs Org-Mode. Database files may thus be conveniently edited using the Org-Mode, see Appendix B.

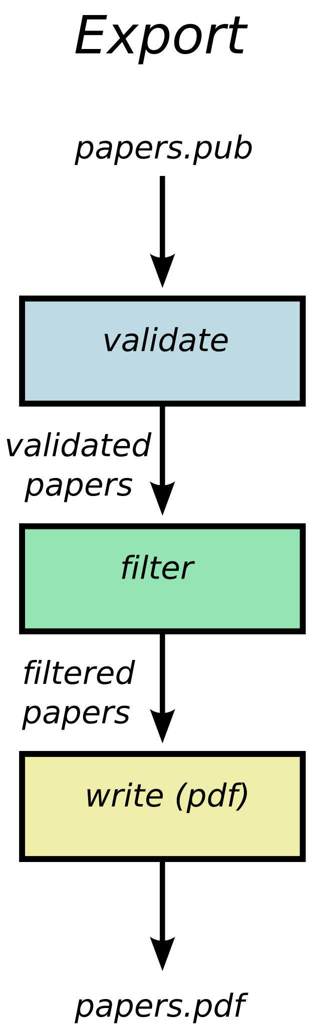


Figure 2.3: Exporting papers.

The papers imported into the system will end up in one of 12 categories, see Table 2.1. When exported to PDF or HTML[2], they will be put under the matching headline.

Category	**Headline**
articles	Articles in International Journals
books	Books
edited	Edited Books
chapters	Chapters in Books
refproceedings	Refereed Proceedings
proceedings	Conference Proceedings
reports	Technical Reports
manuals	Manuals
theses	Theses
courses	Courses
talks	Talks
misc	Other Publications

Table 2.1: Paper categories and headlines.

Which category is assigned to a paper depends on what BibTeX entry-type the paper has, or, alternatively, the category specified by the user when importing papers.

As can be seen in table 2.2, some categories match several entry-types, and some entry-types match several categories. **Publish** implements some simple rules to deduce which category should be assigned to any given paper with known BibTeX entry-type. This is discussed in more detail in Chapter 4.

[2] HTML-output is not implemented in version 1.0

Category	**BibTeX entry-type**
articles	article
books	book
edited	book, proceedings
chapters	inbook
proceedings	inproceedings, conference
reports	techreport
manuals	manual
theses	phdthesis, mastersthesis
courses, talks, misc	misc
not supported	booklet, incollection, unpublished

Table 2.2: Paper categories and corresponding BibTeX entry-types.

Chapter 3

Basic Usage

This chapter contains installation instructions. It also contains basic information on how to use the command-line interface.

3.1 Installation

Publish follows the standard installation procedure for Python packages. Enter the source directory of **Publish** and issue the following command:

```
# sudo python setup.py install
```

See Appendix A for a more detailed description.

3.2 Command-Line Interface

The synopsis for the **Publish** command-line interface looks as follows:

```
# publish <command> [command/global option]... [filename]
```

The command always starts with `publish` followed by a `<command>` which may be be either `import`, `validate`, or `export`. Import and export can be used with some global option, such as `year=2008`, or `author=name_of_choice`. Import and export always require a filename to import from or export to.

Examples:

```
# publish import category=books bookfile.bib

# publish export year=2002 output.pdf
```

The command `validate` can be used by itself or with a file with suffix `.pub`. If used by itself, it is the default database-file `papers.pub` that is validated.

Examples:

```
# publish validate

# publish validate inputfile.pub
```

The command `export` may be used to export a publication record, for example:

```
# publish export papers.pdf
```

This will be discussed in more detail below in Chapters 4 (importing), 5 (validation), and 6 (exporting).

3.3 Global Options

The following global options are recognized by all three commands:

- `debug=yes`

  Enable debugging output. With this option, Python exceptions are not caught, resulting in a full Traceback on errors.

- `autofix=yes`

  Automatically choose the default option when an error is encountered. This is useful (but dangerous) when importing large amounts of data into the database.

3.4 Configuration of the System

Much of the behavior of the system can be configured by the user. The file `publish/config/defaults.py` contains default values of all the variables that can be configured. This includes, for example, the name of the default database file, the venues-file, and the file where invalid papers are stored. It is also possible to change how picky the system is when judging the closeness between papers that are considered as duplicates (`matching_distance_strong`) and how far the system will look to suggest a venue name when a venue (journal name) is not recognized (`matching_distance_weak`).

To change the default settings, the user has to make a file `publish_config.py` and make sure the `PYTHONPATH` variable is set such that any `import publish_config` imports the desired file. This means that the user may have different `publish_config.py` files for different purposes, and the one to be used by **Publish** is controlled by the `PYTHONPATH` variable. In most cases, one would like to change the default database name or other variables for a particular project, and then it is sufficient to make a local `publish_config.py` file in the directory where **Publish** is supposed to be run from and where the database is to be stored (there is no need to set PYTHONPATH in such cases since Python will always try to import from the current working directory first). Note that the user's configuration file *must* have the name publish_config.py.

Suppose we want to change the names of the database and the venues list. For this purpose we make a publish_config.py file with the following content

```
from publish.config.defaults import *

database_filename = ’.publish_papers.pub’
local_venues_filename = ’.publish_venues.txt’
```

Note that we start the filenames with a dot to make them hidden in the project directory.

Sometimes you may want to add words that need special capitalization. Looking at the publish/config/defaults.py file, we realize that such words are stored in a dictionary uppercase. We want to add capitalization of the words "FEniCS" and "FEniCS-based". This means that if those words, however they are capitalized in a title, will be guaranteed to be written that way. The above file will now look as follows

```
from publish.config.defaults import *

database_filename = ’.publish_papers.pub’
local_venues_filename = ’.publish_venues.txt’

uppercase.update({
    "fenics": "{FEniCS}",
    "fenics-based": "{FEniCS}-based",
    })
```

The syntax for specifying capitalization is taken from BibTeX. Note that the key in the uppercase dictionary must be in lowercase. Also note that we do uppercase.update to add new items to the dictionary. Doing uppercase = ... will just assign two entries and forget about all the entries that the default settings take care (i.e., we overwrite the uppercase dictionary imported in the first line). Lists and tuples must be *extended* with new values, typically by an assignment like obj = obj + users_list.

It is possible to skip the first import line in publish_config.py files. **Publish** will then merge the publish_config.py contents with the default values in publish/config/defaults.py. The syntax changes a little bit in publish_config.py since the variables to be configured do not exist from an import. For example, the uppercase variable set above is not imported so we just do an ordinary assignment:

```
database_filename = '.publish_papers.pub'
local_venues_filename = '.publish_venues.txt'

uppercase = {
    "fenics": "{FEniCS}",
    "fenics-based": "{FEniCS}-based",
    }
```

Lists and tuples are not extended; just assign the desired new values.

Chapter 4

Importing

4.1 Overview

Papers may be imported into the system using the `import` command (here for a file in BibTeX format):

```
# publish import inputfile.bib
```

Papers imported to the system will either end up in the database-file (`papers.pub`) or in a file with invalid papers that didn't pass the first round of validation. The file with the invalid papers is saved with date and time so that it can easily be located if needed. During import, the papers will be merged together with the papers already present in the database with particular attention to removing duplicate entries. The database is always saved in the `pub` format.

The following steps are carried out during import (see also Figure 2.1):

1. Read `papers.pub`
2. Validate `papers.pub`, see Chapter 5
3. Read the file which is being imported

4. Validate the imported file, see Chapter 5
5. Merge `papers.pub` with the imported file
6. Make a backup-copy of the database named `papers.pub.bak`
7. Save merge papers to `papers.pub`

4.2 Supported File Formats

In Table 4.1, the mapping between filename suffix and file format is shown. These are the formats currently supported by **Publish**:

Filename Suffix	**File Format**
`.pdf`	PDF
`.bib`	BibTeX
`.bibtex`	BibTeX
`.tex`	LaTeX
`.pub`	`pub`

Table 4.1: Mapping from filename suffix to format.

4.3 Handling Invalid Papers

All invalid papers will automatically be saved to a file postfixed by the current date and time (`invalid_papers-%Y%m%d-%H:%m:%S.pub`), such as for example `invalid_papers-20081128-21:11:47.pub`.

The following steps must typically be carried out when importing papers from a file containing one or more invalid papers:

1. Import file
2. Open file with invalid papers, edit and correct the inaccuracies

3. Import the file with the invalid papers
4. Repeat as needed

Alternatively, one may also edit and validate (see next chapter) the file containing the invalid papers before they are imported into the database.

It is possible to import a file using the option `autofix`:

```
# publish import autofix=yes inputfile.bib
```

When autofix is used, the system automatically answers all questions that arise during validation and merging. This may be useful (but somewhat dangerous) when importing a large number of papers into the database.

To see which inaccuracies that may occur during validation, see Chapter 5 (validation) below.

4.4 Handling Duplicate Papers

If two authors have entered the same paper in different (but similar) ways (for example by specifying the author list in different order) or if a conference proceeding and journal paper have identical names, a duplicate is reported during merging. The system detects this by computing the distance between paper venues (journal names) and paper titles. If this distance between two papers is less than a certain matching distance[1] then the two papers are considered as duplicates.

If a duplicate occurs, the user is faced with the following question: Attribute ”*name of attribute*” differs, what should I do?

1. Keep both papers (marking them as allowed duplicates)

[1] The matching distance `matching_distance_strong` can be edited in `publish/config/general.py`.

2. Ignore papers (marking them as invalid)

3. Keep first paper (*name of first paper*) and ignore second paper (*name of second paper*)

4. Keep second paper (*name of second paper*) and ignore first paper (*name of first paper*)

5. Use attribute from first paper (*value of attribute*)

6. Use attribute from second paper (*value of attribute*)

7. Print diff

If the user chooses to keep both papers, they are both marked as allowed duplicates, and will thereafter not be up for questioning again. If the user instead chooses to ignore the papers, they will both be marked as invalid, and end up in the file with invalid papers. Number three to six in the list are self explanatory. If the user chooses to print a diff, both papers will be shown at the same time in the terminal window, and the differences will be marked, so that the differences are easily spotted for the user.

4.5 Overriding Attributes

When importing a file with papers, it can be useful to specify certain attributes, for example by specifying `department=sc` (to specify publications published by the scientific computing department) and/or `year=2005` (to specify that all papers were published in 2005). This means that all papers imported at this time will get those attributes and later, if one wants to extract only those papers written in 2005, they are easily found. (See Chapter 6 (Exporting) for more information about paper extraction/filtering.) The following command illustrates the above example:

```
# publish import department=sc year=2005 inputfile.bib
```

Overriding attributes is mostly useful when the category is not easily deduced from the BibTeX entry-type. As shown above in Table 2.2, the category is not always evident for each BibTeX entry-type. **Publish** tries to make an intelligent choice, but this is not always possible.

For example, if a paper has the BibTeX entry-type `book`, then **Publish** looks for the attribute `editor`. If there is such an attribute, the `book` will end up in the category `edited`, and if not, the `book` will end up in `books`. Similarly, papers with entry-type `misc` will end up in the category `courses` if the attribute `code` is found, in `talks` if the attribute `meeting` is found, or else in `misc`.

However, **Publish** is currently not able to deduce whether a paper with entry-types `proceedings` or `conference` should be categorized into `proceedings` or `refproceedings`.

This must then be specified explicitly when importing papers:

```
# publish import category=refproceedings inputfile.bib
```

If not otherwise specified, **Publish** places all papers with entry-type `proceeding` (or `conference`) into the category `proceedings`.

Chapter 5

Validation

5.1 Overview

Papers will automatically be validated when imported into the system (including papers already present in the database), and when exported from the system. Validation can also be performed separately by the following command:

```
# publish validate
```

This validates all papers found in the database (`papers.pub`).

One may also validate papers in other files, which must then be stored in the `pub` format:

```
# publish validate inputfile.pub
```

If a paper is valid, it is saved back to the database. Otherwise, it is marked as invalid and saved to a separate file storing invalid papers.

5.2 Checks Performed

Publish checks each paper for a number of common errors:

- Status (status specified)
- Attributes (no required attributes missing)
- Venues (matching against known venues)
- Titles (formatting, capitalization)
- Page range (formatting)
- Typos (some common typos)

We discuss each of these checks below.

5.2.1 Status

All papers must be marked with a status. If the status is missing, it will automatically be set to `published`. The following values should typically be used:

- `inpreparation`
- `submitted`
- `accepted`
- `published`
- `withdrawn`
- `rejected`

5.2.2 Attributes

For each category, a number of attributes are required. For all categories, the attributes `title` and `author` are always required. For a list of which attributes are required for which category, refer to Appendix C.

5.2.3 Venues

The venue (journal, conference, booktitle, etc.) for each paper is validated against a database of known venues. If an exact match is not found, the system tries to find a close match and suggests a correction.

If not found, or if the suggested choice is incorrect, the user may choose to accept the venue name as is. It will then be added to a local file of known venues (`venues.list`), which will be used for subsequent validations.

5.2.4 Titles

Paper titles are validated for correct capitalization. The system keeps a list of common words ("and", "or", "for" etc.) that should not be capitalized, and also a list of words that should be capitalized. The default words for capitalization are set in the `lowercase` and `uppercase` dictionaries. The default configuration is found in `publish/config/defaults.py` and can be extended in a user's `publish_config.py` file.

5.2.5 Page Range

Page ranges must be formatted `x--y`. Illegally formatted page ranges are automatically detected and fixed.

5.2.6 Typos

A number of typos are also detected. The default typos are found in `publish/config/defaults` and can be extended in a user's `publish_config.py` file.

Chapter 6

Exporting

6.1 Overview

Publication records may be exported from the system using the `export` command (here to a file in PDF format):

```
# publish export outputfile.pdf
```

One or more filters may also be specified to extract only papers matching (or not matching) a number of attributes, for example:

```
# publish export year=2008 category=talks \
  author=langtangen outputfile.pdf
```

During export, all papers are read from the database (`papers.pub`), validated, filtered, and then formatted in the desired output format as determined by the filename suffix (see Figure 2.3).

6.2 Supported File Formats

Publish currently exports to the following file formats:

- `pub`
- BibTeX
- LaTeX
- PDF

The LaTeXformat may be useful for generating publication records for inclusion in a CV or an annual report.

6.3 Filtering

Filters may be specified in one of two ways. Either, one may specify an attribute that must be satisfied by all papers to be exported, or one may specify an attribute that *must not* be satisfied by the papers to be exported. It is also possible to list several attributes and mix the two types of filters. The exported papers must then satisfy each one of the listed filters.

The following example demonstrates how to export all journal papers from the year 2008 which are *not* written by H. P. Langtangen:

```
# publish export year=2008 category=articles \
  author!="H. P. Langtangen" papers.pdf
```

When filtering on author names, it is possible to either list only the author's last name, the name with initials, or the full name. Thus, the following examples are all valid, but they may or may not give the same result, depending on how many authors are named "Langtangen", whether or not papers have been entered only with initials or with full names:

```
# publish export author=langtangen papers.pdf
# publish export author="H. P. Langtangen" papers.pdf
# publish export author="Hans Petter Langtangen" papers.pdf
```

In all of these examples, the filter will match all papers where the given author name matches at least one author in the author list (for papers with more than one author).

6.4 Options

The following additional options may be given to the `export` command:

- `view_pdf=no`

  Don't display generated PDF files when exporting to PDF.

- `pdf_viewer=<command>`

  Specify command that should be used to display PDF files.

Appendix A

Installation

Publish follows the standard installation procedure for Python packages. Enter the source directory of **Publish** and issue the following command:

```
# sudo python setup.py install
```

This will install the **Publish** in a subdirectory called `publish` in the default location for user-installed Python packages (usually something like /usr/lib/python2.5/site-packages). In addition, the main script `publish` will be installed in the default directory for user-installed Python scripts (usually in /usr/bin). To see a list of optional parameters to the installation script, type

```
# python setup.py --help
```

Appendix B

Emacs Org-Mode

Emacs Org-Mode makes it convinient to edit the files that are saved in the internal pub format. Emacs Org-Mode folds and unfolds sections of a file indicated with one or more * at the press of the TAB key.

Thus, one may easily fold/unfold paper categories and paper attributes which simplifies editing of large publication database files.

For download and installation instructions, visit http://orgmode.org/.

To activate Org-Mode when .pub files are opened, the following lines should be added to the user’s .emacs file:

```
(setq auto-mode-alist
      (append ’(("\\.org$" . org-mode))
              ’(("\\.pub$" . org-mode))
              auto-mode-alist))
```

Appendix C

Internal Format (.pub)

C.1 Overview

Papers are stored in the pub format according to the following scheme:

```
* category
** title
   attribute: value
   attribute: value
   ...
** title
   attribute: value
   attribute: value
   ...
* category
   ...
```

Each category is specified by a * followed by a single space and the name of the category. Within each category, each paper is specified by ** followed by a single space and the title of the paper. Then follows for each paper all attribute-value pairs separated by : (spaces ignored).

Each category specifies a number of required attributes that must be specified for each paper within that category. In addition to the required attributes, any number of additional attributes may be specified.

Currently, all attribute-value pairs must be entered in a single line (multi-line values not supported).

C.2 Example

```
* articles
** Numerical Methods for Incompressible Viscous Flow
   author:    H. P. Langtangen, K.-A. Mardal, R. Winther
   journal:   Advances in Water Resources
   year:      2002
   status:    published
   number:    8
   pages:     1125--1146
   volume:    25
   key:       SC.4.Langtangen.2002.b
   entrytype: article
** Unified Framework for Finite Element Assembly
   author:    Martin Sandve Aln{\ae}s, Anders Logg, Kent-Andre Mardal, Ola Skavhaug, Hans Petter Langtangen
   journal:   International Journal of Nonlinear Modelling in Science and Engineering
   year:      2009
   status:    published
   entrytype: article
   key:       Simula.SC.96
* books
** Computational Partial Differential Equations - Numerical Methods and Diffpack Programming
   author:    H. P. Langtangen
   publisher: Springer-Verlag
   year:      2003
   status:    published
   note:      2nd edition, 855 pages
   key:       SC.1.Langtangen.2003
   entrytype: book
** {P}ython Scripting for Computational Science
   author:    Hans Petter Langtangen
   publisher: Springer-Verlag
   year:      2008
   status:    published
   isbn:      978-3-540-73915-9
   edition:   third
   duplicate: True
   key:       Simula.SC.63
   entrytype: book
   address:   Heidelberg
* theses
** Applications of High Level Software for Parallel Solution of Partial Differential Equations
   author:     {\AA}smund {\O}deg{\aa}rd
   school:     Department of Informatics, University of Oslo
   year:       2006
   thesistype: phd
   status:     published
   key:        Odegard.2006.1
   entrytype:  phdthesis
```

C.3 Required Attributes

Some attributes are mandatory for a given category. For example, an article needs author, title, journal, and year (as well as status, but that can set by **Publish**()). The required attributes are defined in a variable `category_attributes` in `publish/config/defaults.py` and can be overridden in a user's `publish_config.py` file.

We refer to the `category_attributes` dictionary in `publish/config/defaults.py` for an overview of all required attributes.

Appendix D

Venues

Below are listed all venues currently known by the system.

D.1 Journals

1. none
   none
   ISSN: xxxx-xxxx

2. ACM Transactions on Mathematical Software
   ACM Trans. Math. Software
   ISSN: 0098-3500

3. ACM Computing Surveys
   ACM Comput. Surv.
   ISSN: 0360-0300

4. ACM SIGSOFT Software Engineering Notes
   ACM SIGSOFT Software Engineering Notes
   ISSN: 01635848

5. ACM Transactions on Multimedia Computing, Communications, and Applications
   ACM Trans. Multimed. Comput. Comm. Appl.
   ISSN: 1551-6857

6. ACM Transactions on Software Engineering and Methodology
   ACM Trans. Software Eng. Meth.
   ISSN: 1049-331X

7. Advances in Computational Mathematics
   Adv. Comput. Math.
   ISSN: 1019-7168
8. Advances in Water Resources
   Adv. Water Resour.
   ISSN: 0309-1708
9. Annals of Biomedical Engineering
   Ann. Biomed. Eng.
   ISSN: 0090-6964
10. Applied Cognitive Psychology
    Appl. Cognit. Psychol.
    ISSN: 0888-4080
11. Applied Mathematics and Computation
    Appl. Math. Comput.
    ISSN: 0096-3003
12. Applied Mathematical Finance
    Appl. Math. Finance
    ISSN: 1350-486X
13. Applied Mathematical Modelling
    Appl. Math. Model.
    ISSN: 0307-904X
14. Applied Mechanics and Engineering
    Appl. Mech. Eng.
    ISSN: 1425-1655
15. Archives of Computational Methods in Engineering
    Arch. Comput. Methods Eng.
    ISSN: 1134-3060
16. Behaviour & Information Technology
    Behaviour & Information Technology
    ISSN: 0144-929X
17. Biophysical Journal
    Biophys. J.
    ISSN: 0006-3495
18. BIT Numerical Mathematics
    BIT
    ISSN: 0006-3835
19. Campus-Wide Information Systems
    Camput-Wide Information Systems
    ISSN: 1065-0741

20. Computers in Biology and Medicine
    Comput. Biol. Med.
    ISSN: 0010-4825

21. Computer Communications
    Comput. Comm.
    ISSN: 0140-3664

22. Computational Geosciences
    Comput. Geosci.
    ISSN: 1420-0597

23. Communications in Computational Physics
    Comput. Phys.
    ISSN: 1814-2406

24. Computers & Mathematics with Applications
    Comput. Math. Appl.
    ISSN: 0898-1221

25. Computational Mechanics. Solids, Fluids, Engineered Material, Aging Infrastructure, Molecular Dynamics, Heat Transfer, Manufacturing Processes, Optimization, Fracture & Integrity
    Comput. Mech.
    ISSN: 0178-7675

26. Computer Methods in Applied Mechanics and Engineering
    Comput. Methods Appl. Mech. Engrg
    ISSN: 0045-7825

27. Computer Methods in Biomechanics and Biomedical Engineering
    Comput. Meth. Biomech. Biomed. Eng.
    ISSN: 1025-5842

28. Computing in Science & Engineering
    Comput. Sci. Eng.
    ISSN: 1521-9615

29. Computing and Visualization in Science
    Comput. Vis. Sci.
    ISSN: 1432-9360

30. Continental Shelf Research
    Continent. Shelf Res.
    ISSN: 0278-4343

31. Empirical Software Engineering
    Empir. Software Eng.
    ISSN: 1382-3256

32. Engineering Analysis with Boundary Elements
    Eng. Anal. Bound. Elem.
    ISSN: 0955-7997

33. Formal Aspects of Computing
    Formal Aspect. Comput.
    ISSN: 0934-5043

34. Future Generation Computer Systems
    Future Generat. Comput. Syst.
    ISSN: 0167-739X

35. Global Journal of Pure and Applied Mathematics
    Global J. Pure Appl. Math.
    ISSN: 0973-1768

36. IEEE/ACM Transactions on Networking
    IEEE/ACM Transactions on Networking
    ISSN: 1063-6692

37. IEEE Communications Letters
    IEEE Comm. Lett.
    ISSN: 1089-7798

38. IEEE Communications Magazine
    IEEE Comm. Mag.
    ISSN: 0163-6804

39. IEEE Computer Applications in Power Magazine
    IEEE Comput. Appl. Power Mag.
    ISSN: 0895-0156

40. IEEE Computer Architecture Letters
    IEEE Comput. Architect. Lett.
    ISSN: 1556-6056

41. IEEE Control Systems Magazine
    IEEE Control Syst. Mag.
    ISSN: 1066-033X

42. IEEE Distributed Systems Online
    IEEE Distr. Syst. Online
    ISSN: 1541-4922

43. IEEE Micro Magazine
    IEEE Micro Magazine
    ISSN: 0272-1732

44. IEEE Network: The Magazine of Global Information Exchange
    IEEE Netw. Mag. Global Inform. Exchange
    ISSN: 0890-8044

45. IEEE Software Magazine
    IEEE Software Mag.
    ISSN: 0740-7459

46. IEEE Transactions on Biomedical Engineering
    IEEE Trans. Biomed. Eng.
    ISSN: 0018-9294

47. IEEE Transactions on Computers
    IEEE Trans. Comput.
    ISSN: 0018-9340

48. IEEE Transactions on Industrial Informatics
    IEEE Transactions on Industrial Informatics
    ISSN: 1551-3203

49. IEEE Transactions on Parallel and Distributed Systems
    IEEE Trans. Parallel Distr. Syst.
    ISSN: 1045-9219

50. IEEE Transactions on Software Engineering
    IEEE Trans. Software Eng.
    ISSN: 0098-5589

51. IEEE Transactions on Wireless Communications
    IEEE Trans. Wireless Comm.
    ISSN: 1536-1276

52. IET Communications
    IET Communications
    ISSN: 1751-8628

53. IMA Journal of Applied Mathematics
    IMA J. Appl. Math.
    ISSN: 0272-4960

54. IMA Journal of Numerical Analysis
    IMA J. Numer. Anal.
    ISSN: 0272-4979

55. Information Processing Letters
    Inform. Process. Lett.
    ISSN: 0020-0190

56. Information and Software Technology
    Inform. Software Tech.
    ISSN: 0950-5849

57. ISA Transactions
    ISA Transactions
    ISSN: 0019-0578

58. International Journal of Applied Mechanics and Engineering
    Appl. Mech. Eng.
    ISSN: 1425-1655

59. International Journal for Computational Methods in Engineering Science and Mechanics
    J. Comput. Methods Eng. Sci. Mech.
    ISSN: 1550-2287

60. International Journal of Autonomous and Adaptive Communications Systems
    International Journal of Autonomous and Adaptive Communications Systems
    ISSN: 1754-8632

61. International Journal of Bioelectromagnetism
    International Journal of Bioelectromagnetism
    ISSN: 1456-7857

62. International Journal of Computational Engineering Science
    Int. J. Comput. Eng. Sci.
    ISSN: 1465-8763

63. International Journal of Computational Science and Engineering
    IJCSE
    ISSN: 1742-7185

64. International Journal of Computer Mathematics
    Int. J. Comput. Math.
    ISSN: 0020-7160

65. International Journal of Forecasting
    Int. J. Forecast
    ISSN: 0169-2070

66. International Journal of Fluid Mechanics Research
    Int. J. Fluid Mech. Res.
    ISSN: 1064-2277

67. International Journal of Numerical Analysis and Modeling
    Int. J. Numer. Anal. Model.
    ISSN: 1705-5105

68. International Journal of Nonlinear Sciences and Numerical Simulation
    Int. J. Nonlinear Sci. Numer. Simul.
    ISSN: 1565-1339

69. International Journal of Nonlinear Modelling in Science and Engineering
    Int. J. Nonlinear Model. Sci. Eng.
    ISSN: 1472-085X

70. International Journal for Numerical Methods in Engineering
    Int. J. Numer. Meth. Eng.
    ISSN: 0029-5981

71. International Journal for Numerical Methods in Fluids
    Int. J. Numer. Meth. Fluid.
    ISSN: 0271-2091

72. International Journal of Project Management
    Int. J. Proj. Manag.
    ISSN: 0263-7863

73. International Journal of Pure and Applied Mathematics
    International Journal of Pure and Applied Mathematics
    ISSN: 1311-8080

74. International Journal of Software Engineering and Knowledge Engineering
    Int. J. Software Eng. Knowl. Eng.
    ISSN: 0218-1940

75. Journal of Computational Acoustics
    J. Comput. Acoust
    ISSN: 0218-396X

76. Journal of Computational and Applied Mathematics
    J. Comput. Appl. Math.
    ISSN: 0377-0427

77. Journal of Computational Finance
    J. Comput. Finance
    ISSN: 1460-1559

78. Journal of Computational Neuroscience
    J. Comput. Neurosci.
    ISSN: 0929-5313

79. Journal of Computational Physics
    J. Comput. Phys.
    ISSN: 0021-9991

80. Journal d'Analyse Mathematique
    Journal d'Analyse Mathematique
    ISSN: 0021-7670

81. Journal of Geodynamics
    J. Geodyn.
    ISSN: 0264-3707

82. Journal of Geophysical Research (JGR)-B
    J. Geophys. Res. B
    ISSN: 0196-6936

83. Journal of Medical Ultrasonics
    Journal of Medical Ultrasonics
    ISSN: 1346-4523

84. Journal of Network and Computer Applications
    J. Netw. Comput. Appl.
    ISSN: 1084-8045

85. Journal of Numerical Mathematics
    J. Numer. Math.
    ISSN: 1570-2820

86. Journal of Object Technology
    J. Object Tech.
    ISSN: 1660-1769

87. Journal of Scientific Computing
    J. Sci. Comput.
    ISSN: 0885-7474

88. Journal of Software Maintenance and Evolution: Research and Practice
    J. Software Mainten. Evol. Res. Pract.
    ISSN: 1532-060X

89. Journal of Statistical Software
    J. Stat. Software
    ISSN: 1548-7660

90. Journal of Systems Architecture
    J. Syst. Architect.
    ISSN: 1383-7621

91. Journal of Systems and Software
    J. Syst. Software
    ISSN: 0164-1212

92. Mathematical Biosciences
    Math. Biosci.
    ISSN: 0025-5564

93. Mathematics of Computation
    Math. Comp.
    ISSN: 0025-5718

94. Mathematics and Computers in Simulation
    Math. Comput. Simulat.
    ISSN: 0378-4754

95. Mathematical Geology
    Math. Geol.
    ISSN: 0882-8121

96. Metheoritics & Planetary Science
    Metheoritics Planet. Sci.
    ISSN: 1086-9379

97. Mobile Networks and Applications
    Mobile Network. Appl.
    ISSN: 1383-469X

98. Modeling, Identification and Control. A Norwegian Research Bulletin
    Model. Identif. Control
    ISSN: 0332-7353

99. Multi Agent and Grid Systems
    Multi Agent and Grid Systems
    ISSN: 1574-1702

100. Nordic Journal of Computing
    Nord. J. Comput.
    ISSN: 1236-6064

101. Norwegian Journal of Geography/ Norsk Geografisk Tidsskrift
    Norwegian Journal of Geography/ Norsk Geografisk Tidsskrift
    ISSN: 0029-1951

102. Numerical Linear Algebra with Applications
    Numer. Linear Algebra Appl.
    ISSN: 1070-5325

103. Numerical Methods for Partial Differential Equations. An International Journal
    Numer. Methods Partial Differential Equations
    ISSN: 0749-159X

104. Numerische Mathematik
    Numer. Math.
    ISSN: 0029-599X

105. Optics Communications
    Otic. Comm.
    ISSN: 0030-4018

106. Performance Evaluation
    Perform. Eval.
    ISSN: 0166-5316

107. Personal and Ubiquitous Computing
    Personal and Ubiquitous Computing
    ISSN: 1617-4909

108. Physics of the Earth and Planetary Interiors
    Phys. Earth Planet. In.
    ISSN: 0031-9201

109. Physical Review B: Condensed Matter and Materials Physics
    Phys. Rev. B Condens. Matter
    ISSN: 1098-0121

110. Proceedings / IEEE Computer Society Bioinformatics Conference. IEEE Computer Society Bioinformatics Conference
Proc. IEEE Comput. Soc. Bioinformatics Conf. IEEE Comput. Soc. Bioinformatic
ISSN: 1555-3930

111. Research Letters in Communications
Research Letters in Communications
ISSN: 1687-6741

112. Scandinavian Journal of Information Systems
Scand. J. Inform. Syst.
ISSN: 0905-0167

113. Scientific Programming
Sci. Program.
ISSN: 1058-9244

114. SIAM Journal on Numerical Analysis
SIAM J. Numer. Anal.
ISSN: 0036-1429

115. SIAM Journal on Scientific Computing
SIAM J. Sci. Comput.
ISSN: 1064-8275

116. Software Process: Improvement and Practice
Software Process Improv. Pract.
ISSN: 1077-4866

117. Software and Systems Modeling
Software Syst. Model.
ISSN: 1619-1366

118. Stroke
Stroke
ISSN: 0039-2499

119. Transactions on Machine Learning and Data Mining
Transactions on Machine Learning and Data Mining
ISSN: 1865-6781

120. Transport in Porous Media
Transp. Porous Media
ISSN: 0169-3913

121. Wireless Personal Communications
Wireless Pers. Comm.
ISSN: 0929-6212

D.2 Publishers

1. Birkhäuser
2. Cambridge University Press
3. Chapman and Hall
4. Computers in Cardiology
5. Elsevier Science
6. EuroPACS
7. European Association of Geoscientists & Engineers
8. European Community on Computational Methods in Applied Sciences
9. Informing Science Press
10. John von Neumann Institute for Computing
11. Kluwer Academic Publishers
12. Literatür Yayincilik Ltd.
13. MGNet
14. Nationellt centrum för matematikutbildning, NCM
15. North Holland
16. Nova Science Publishers
17. SIAM
18. SPIE

19. Springer

20. Swansea

21. Tapir Academic Press

22. Taylor & Francis Books Ltd

23. Wiley Press

D.3 Schools

1. Department of Informatics, University of Oslo

2. Faculty of Mathematics and Natural Sciences, University of Oslo

D.4 Institutions

1. Carleton University

2. Finite Element Center

3. InterMedia, University of Oslo

4. Norwegian University of Science and Technology

5. Numerical Objects A.S.

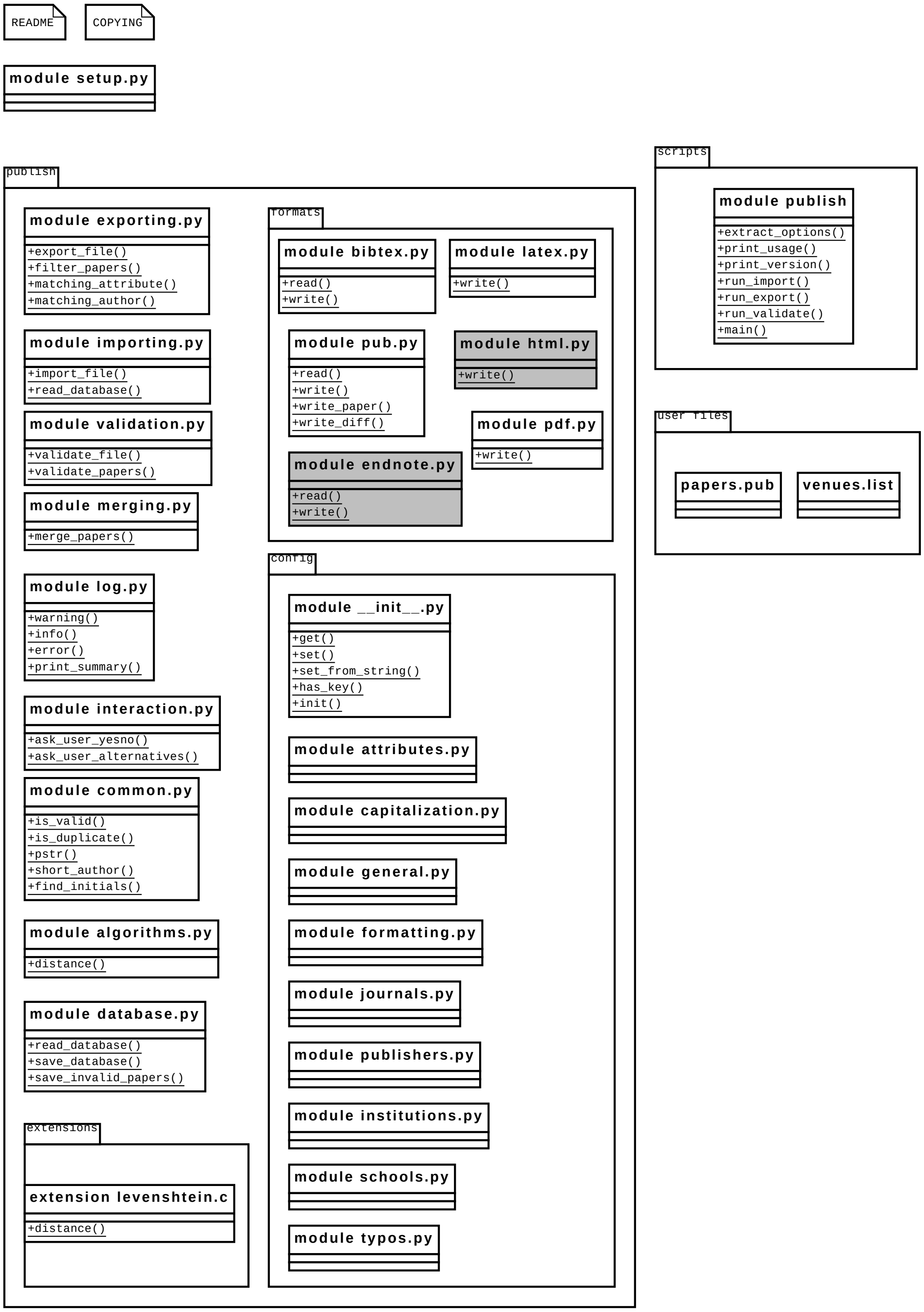
6. University of Castilla-La Mancha

7. Simula Research Laboratory

Appendix E

Design

The figure below illustrates the design of **Publish**. Modules that are coloured grey have not yet been implemented but may be added in future versions of **Publish**.


README
COPYING
module setup.py
publish
module exporting.py
+export_file()
+filter_papers()
+matching_attribute()
+matching_author()
module importing.py
+import_file()
+read_database()
module validation.py
+validate_file()
+validate_papers()
module merging.py
+merge_papers()
module log.py
+warning()
+info()
+error()
+print_summary()
module interaction.py
+ask_user_yesno()
+ask_user_alternatives()
module common.py
+is_valid()
+is_duplicate()
+pstr()
+short_author()
+find_initials()
module algorithms.py
+distance()
module database.py
+read_database()
+save_database()
+save_invalid_papers()
extensions
extension levenshtein.c
+distance()
formats
module bibtex.py
+read()
+write()
module latex.py
+write()
module pub.py
+read()
+write()
+write_paper()
+write_diff()
module html.py
+write()
module pdf.py
+write()
module endnote.py
+read()
+write()
config
module __init__.py
+get()
+set()
+set_from_string()
+has_key()
+init()
module attributes.py
module capitalization.py
module general.py
module formatting.py
module journals.py
module publishers.py
module institutions.py
module schools.py
module typos.py
scripts
module publish
+extract_options()
+print_usage()
+print_version()
+run_import()
+run_export()
+run_validate()
+main()
user files
papers.pub
venues.list

Appendix F

License

Publish is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

The GNU GPL is included verbatim below.

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Preamble

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

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

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

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

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

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

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

TERMS AND CONDITIONS

0. Definitions.

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

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

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

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

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

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

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

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

1. Source Code.

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

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

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

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

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

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

2. Basic Permissions.

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

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

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

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

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

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

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

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

5. Conveying Modified Source Versions.

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

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

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

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

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

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

6. Conveying Non-Source Forms.

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

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

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

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

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

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

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

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

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

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

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

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

7. Additional Terms.

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

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

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

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

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

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

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

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

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

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

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

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

8. Termination.

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

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

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

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

9. Acceptance Not Required for Having Copies.

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

10. Automatic Licensing of Downstream Recipients.

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

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

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

11. Patents.

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

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

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

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

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

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

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

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

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

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

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

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

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

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

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

THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

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

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

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

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

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

    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

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

    You should have received a copy of the GNU General Public License
    along with this program.  If not, see <http://www.gnu.org/licenses/>.
```

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

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

```
    <program>  Copyright (C) <year>  <name of author>
    This program comes with ABSOLUTELY NO WARRANTY; for details type ‘show w’.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type ‘show c’ for details.
```

The hypothetical commands ‘show w’ and ‘show c’ should show the appropriate parts of the General Public License. Of course, your program’s commands might be different; for a GUI interface, you would use an "about box".

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

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

Index