

# MakeJmlrBookGUI User Manual

## Version 0.9.6b

Nicola L.C. Talbot

<http://www.dickimaw-books.com/>

16<sup>th</sup> May, 2018

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Installing the jmlr LaTeX Bundle</b>	<b>4</b>
<b>3</b>	<b>Main Window</b>	<b>5</b>
<b>4</b>	<b>Creating a New Book</b>	<b>6</b>
<b>5</b>	<b>Loading a Book</b>	<b>21</b>
<b>6</b>	<b>The Build Menu</b>	<b>22</b>
6.1	Building the Main Book PDF . . . . .	22
6.2	Building the Print Version . . . . .	22
6.3	Building Individual Papers . . . . .	22
6.4	Create Proceedings bib File . . . . .	23
6.5	Create Proceedings zip File . . . . .	23
<b>7</b>	<b>The Tools Menu</b>	<b>23</b>
7.1	Check Characters Outside the Range 0x20–0x7E . . . . .	23
<b>8</b>	<b>Application Settings</b>	<b>24</b>
<b>9</b>	<b>Troubleshooting</b>	<b>26</b>
9.1	Examples of common problems . . . . .	27
9.2	Error Messages . . . . .	36
	<b>Abbreviations</b>	<b>36</b>
	<b>Index</b>	<b>38</b>
<b>A</b>	<b>Licence</b>	<b>39</b>

## 1 Introduction

MakeJmlrBookGUI is a Java GUI provided to help editors build [Proceedings of Machine Learning Research \(PMLR\)](#) (formerly [Journal of Machine Learning Research Workshop and Conference Proceedings \(JMLR W&CP\)](#)) or books in the [Challenges in Machine Learning \(CiML\)](#) series. This application is designed to only work with books using the  $\text{\LaTeX}$  `jmlrbook` class that import papers using the `jmlr` class. See the `jmlr` documentation for further details on using those classes. This manual assumes that the user is familiar with using  $\text{\LaTeX}$  and the `jmlr` bundle. Make sure you have an up-to-date version of the `jmlr` bundle [installed](#).

It is possible to process all the articles required for the [PMLR](#) manually without using MakeJmlrBookGUI, but if you choose to do so, you must follow all the [PMLR](#) submission guidelines described in the [PMLR Proceedings Specification](#).

In addition to this manual, there are also some video tutorials available at <http://www.dickimaw-books.com/software/makejmlrbookgui/videos/>. Some of the images here and the video tutorials show the old [JMLR W&CP](#) settings. If you are preparing a book with reprints of [JMLR W&CP](#) articles, make sure the class option `wcp` is used, otherwise use `pmlr` for new articles. You will need at least version 1.23 of the `jmlr` and `jmlrbook` class to support this option.

Most of the examples in this document use the file list shown in [Figure 1](#). The files `beer11a.tex`, `darcella.tex`, `doella.tex`, `smith11a.tex` and `winter11a.tex` are individual PMLR papers. The file names used are those required by the PMLR naming conventions (surname of first author followed by two-digit year of publication and iterative letter code for repeated names). Each of these files uses the `jmlr` class and can be compiled as a standalone document.

The file `jmlrbook-sample.tex` is the main book file, which imports each of the standalone articles listed above. This file uses the `jmlrbook` class. It also contains the following lines:

```
\begin{jmlrpapers}

  \importpaper{beer11a}{beer11a}

  \importpaper{darcella}{darcella}

  \importpaper{doella}{doella}

  \importpaper{smith11a}{smith11a}

  \importpaper{winter11a}{winter11a}
```

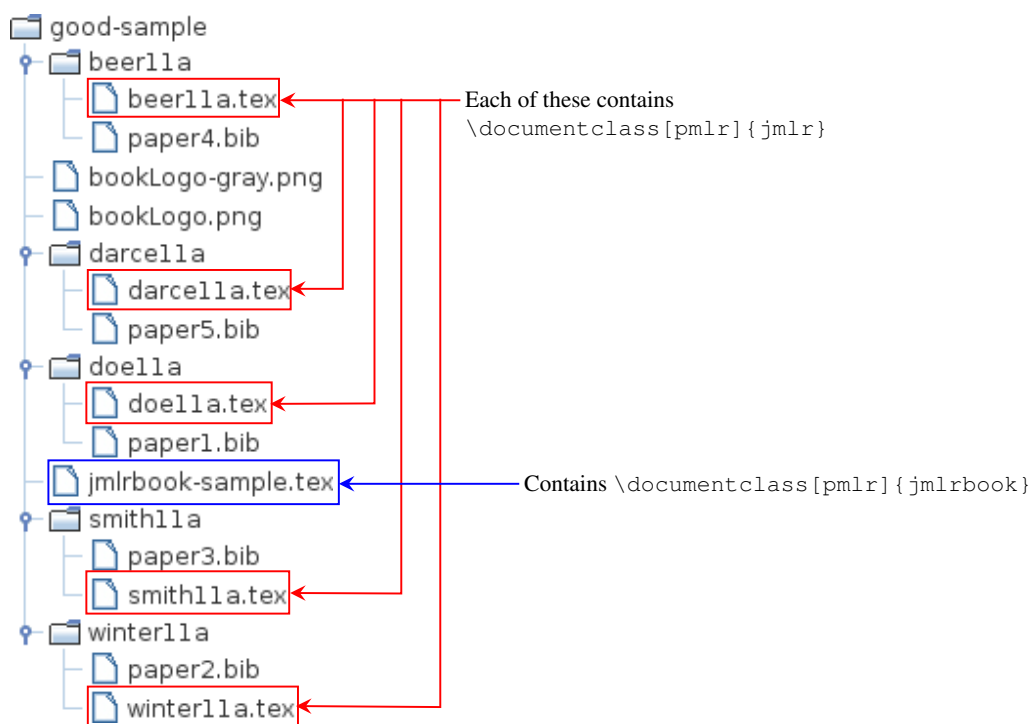


Figure 1: Sample Book File List

```
\end{jmlrpapers}
```

You also need to ensure that the main book file includes all the packages used by the imported articles. If you're wondering how it is that a  $\text{\LaTeX}$  document can import files that each contain a `\documentclass` command, the `jmlrbook` class uses the `combine` class, which makes this possible.

Since it can be rather complicated to set up the main book file, `MakeJmlrBookGUI` has a function that can do this for you. I strongly recommend you use this function to help reduce the chances of complicated errors. See [§4 Creating a New Book](#) for further details.

To start `MakeJmlrBookGUI`, run the command `makejmlrbookgui`. The following options are available:

**-version** Print application version number.

**-help** Print application version number and syntax.

**-debug** Switch on debug mode.

**-nodebug** Switch off debug mode (default).

**-timeout**  $\langle n \rangle$  Set the maximum length of time for processes, where  $\langle n \rangle$  is an integer representing the number of milliseconds until the process times out. (This ensures that if things go very badly wrong and one of the processes doesn't complete, e.g. T<sub>E</sub>X gets stuck in an infinite loop, it will be terminated after the given length of time. Otherwise the process will carry on, even after you've quit MakeJmlrBookGUI.)

Note: you can use sectioning commands, such as `\chapter`, outside of the imported papers (for example, you may want supplemental material in an appendix). *However, `\chapter`, `\appendix` etc must be placed outside of the `jmlrpapers` environment.* For example:

```
\begin{jmlrpapers}

  \importpaper{beer11a}{beer11a}

  \importpaper{darcella}{darcella}

  \importpaper{doe11a}{doe11a}

  \importpaper{smith11a}{smith11a}

  \importpaper{winter11a}{winter11a}

\end{jmlrpapers}

\appendix
\chapter{Datasets}
```

or

```
\begin{jmlrpapers}

  \importpaper{beer11a}{beer11a}

  \importpaper{darcella}{darcella}

  \importpaper{doe11a}{doe11a}

  \importpaper{smith11a}{smith11a}

  \importpaper{winter11a}{winter11a}
```

```
\end{jmlrpapers}

\appendix

\begin{jmlrpapers}
  \importarticle{dataset}{dataset}
\end{jmlrpapers}
```

## 2 Installing the jmlr LaTeX Bundle

This application requires an up-to-date version of the jmlr bundle installed on your computer. **This should be properly installed, not just placed in the same directory as the book.** The `pmlr` class option was added to version 1.23, so you need to make sure you have that version (or a newer one) if this option is required (which it will be for new articles).

If you don't have the jmlr bundle installed, or your version is too old, you can use your T<sub>E</sub>X installation package manager to install or update the class files. For example, with the T<sub>E</sub>X Live update manager:

```
tlmgr update jmlr
```

(See the MikTeX documentation to find out how to do this with MiKTeX.)

This is the recommended method, but if you don't know how to do this or you are otherwise unable to do this, you can use MakeJmlrBookGUI's Tools → Fetch JMLR LaTeX Bundle menu item.

Alternatively, you can install manually:

1. Find out where to install T<sub>E</sub>X bundles on your computer, if you don't already know. To do this, open a command prompt or terminal and type:

```
kpsewhich -var-value=TEXMFHOME
```

This should print the name of the appropriate directory. For example, `~/texmf` or `C:\localtexmf`

2. Fetch the installation bundle from <http://mirror.ctan.org/install/macros/latex/contrib/jmlr.tds.zip>
3. Extract the above ZIP file to the directory given in Step 1. For example, on Unix-like systems:

```
unzip -d ~/texmf ~/Downloads/jmlr.tds.zip
```

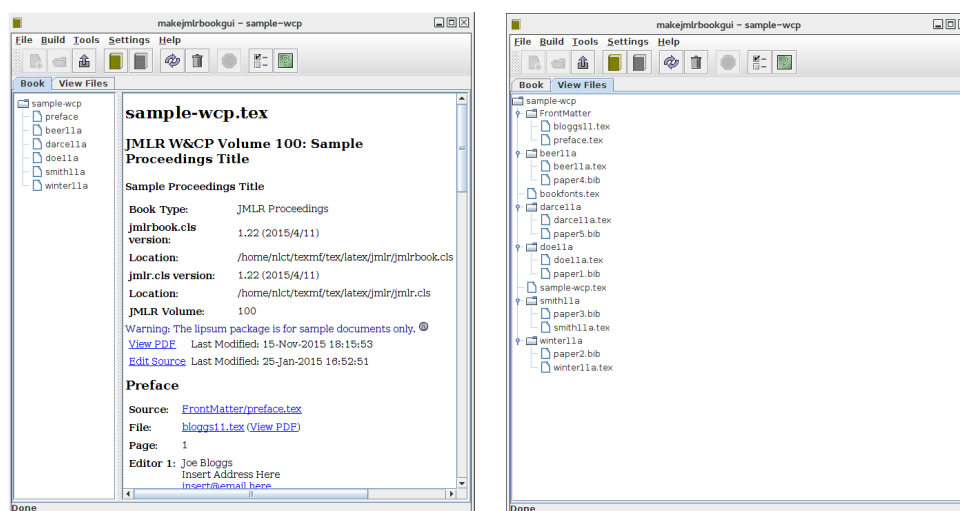
4. Update the TeX database. There should be a command called `texhash` or `mktextlsr`. Run it with the name provided in Step 1. For example:

```
mktextlsr C:\localtexmf
```

or

```
texhash ~/texmf
```

### 3 Main Window



(a)


(b)

Figure 2: The Main Window: (a) the book details visible; (b) the book file list visible.


The main window (Figure 2) has a tool bar, two tabbed panes and a status bar. The tool bar may be repositioned above, below, left or right of the tabbed panes. The status bar is always positioned below the tabbed panes. The tabbed panes consist of:

#### Book Details Tab

Select the book details tab to show information about the book (Figure 2(a)).

This tab has a navigation panel with links to information about the imported articles. The book details include links to edit the source code or view the resulting document files. You can set the applications required to edit text files or view PDF files using the [properties dialog](#). You can save the displayed information using the File → Export Messages... function to export to an HTML file. (The further information buttons  won't work in the exported HTML file.)

There are two types of the complete book PDF that you can create: the colour version with hyperlinks for on-line viewing and the grey version without hyperlinks for printing. The links to these PDF files are identified as “View PDF (On-Line Version)” and “View PDF (Print Version)”. These PDF files are only required for the [CiML](#) books. For [PMLR](#), the colour book PDF is used to obtain all the necessary information to build the individual papers and should not be sent to the proceedings production editor. (Instead, just send the .zip file containing the articles and .bib file as described in [§6.5 Create Proceedings zip File](#).)

The book details tab also lists any errors that were encountered when compiling the book (shown in red) and, if the [strict option](#) is set, any non-conformance detected (shown in purple). (See [§9 Troubleshooting](#).) Warning messages are shown in navy. Many of these messages have a further information icon  that you can click on to open a message window with more detailed information and recommendations. If you prefer to change this icon to the text “[Details]” or if you want both the icon and [Details] displayed, then you can switch to these settings using the Properties dialog, described in [§8 Application Settings](#).

### Book File List

The book file list tab ([Figure 2\(b\)](#)) shows all  $\text{T}_\text{E}\text{X}$ , Bib $\text{T}_\text{E}\text{X}$  and image files in the book directory and subdirectories. You can double-click on one of these files to view them. You can refresh this list using Tools → Refresh Files...

## 4 Creating a New Book

To create a new book, first ensure that you have unpacked all the article source files to a temporary location. (Don't use the new book function until you have *all* the article source files. If new articles need to be added after the new book function has been used, you will have to add the files manually, and adjust the code in the main .tex file as appropriate.)

In this example, I have put the files in ~/Documents/PMLR/OriginalFiles/ listed as follows:



```
~/Documents/PMLR/OriginalFiles/  
  
  MaryJanedeBeer/  
    article.tex  
    references.bib  
    shapes.pdf  
  m-j-darce/  
    figs/  
      shapes.png  
    WorkshopOnSomethingArticle.tex  
    refs.bib  
  DoeJones/  
    DoeJones-suppl.pdf  
    main.tex  
    refs1.bib  
    refs2.bib  
    results.tex  
    table1.tex  
    table2.tex  
  Smith/  
    article.tex  
    jmlr2e.sty  
    refs.bib  
    shapes.eps  
  WFJ/  
    figs/  
      shapes.pdf  
    mystuff.sty  
    refs.bib  
    WinterFinchleyJones.tex
```

There are some commonly-encountered issues with some of these source files:

1. Some of the articles have PDF or PNG images, which indicates that their authors use `pdflatex`, but one of the articles (Smith) has an EPS file, which means that the author used `latex` rather than `pdflatex`.

The original `latex` application (which creates a DVI file) can read EPS files, but it can't read PDF or PNG (or JPG) files. Whereas `pdflatex` can read PDF, PNG and JPG files, but can't read EPS files. When we compile the book, we can't switch output devices during compilation, so this could potentially cause problems.

Luckily most modern  $\text{T}_{\text{E}}\text{X}$  distributions are able to convert EPS to PDF during document compilation, but to ensure this works, the image file should be included in the document using `\includegraphics{<imagefile>}`, where `<imagefile>` is the name of the image file without the file extension.

This is one of the reasons why `MakeJmlrBookGUI` uses `pdflatex` rather than `latex`.<sup>1</sup> Unfortunately, in this example the author has used the obsolete `epsfig` package and has included the image with `\epsfig`. This means that in order for the book to compile correctly, all instances of `\epsfig` must be replaced with `\includegraphics` with the file extension removed. This is a time-consuming task, especially if the author hasn't used a single `.tex` file for the article source, but has spread the document code out in multiple `.tex` files that have been included with `\input`.

`MakeJmlrBookGUI`'s new book function tries to overcome this problem by performing the substitutions for you.

2. Another problem with the Smith article is that it uses the old `jmlr2e` package with the `article` class instead of using just the new `jmlr` class. Since the `jmlrbook` class depends on the imported articles using the `jmlr` class, this will cause a problem when compiling the book.

`MakeJmlrBookGUI`'s new book function can detect this problem and will perform the relevant substitutions.

3. In the `WFJ` article, the authors have provided a custom package (called `mystuff.sty`) loaded in `WinterFinchleyJones.tex`. This typically just contains a series of short cut command definitions. Since this isn't a standard package, the main book file won't find it. While it is possible to just put a copy of `mystuff.sty` in the main book directory, it's possible that the definitions in `mystuff.sty` will conflict with custom commands defined in the other articles. For example, it's quite typical for

---

<sup>1</sup>Another reason is that `pdflatex` handles hyperlinks better than `latex`.

authors to define `\eg` as a short cut for “e.g.” (possibly with space factor adjustments).

When each article is imported into the book, the article source is scoped, so that any definitions made with `\newcommand` or `\renewcommand` only have an effect within the article. This means that if, say, both de Beer et al’s article (`article.tex`) and Doe et al’s article `main.tex` have:

```
\newcommand{\eg}{e.g.\ }
```

there won’t be a conflict. However, if `mystuff.sty` has also defined `\eg` and you include it in the main book file, then `\eg` will already be defined when `article.tex` and `main.tex` try to define it. Therefore it’s not a good idea to use `mystuff.sty` for the book. If the `mystuff` package just contains a series of command definitions, then the best course of action is to just copy those command definitions into the article’s main file (`WinterFinchleyJones.tex` in this case) and remove `\usepackage{mystuff}`.

MakeJmlrBookGUI’s new book function will alert you if an article tries to load a package that is not in your T<sub>E</sub>X distribution.

4. The d’Arçé (`m-j-darce`) article has a subdirectory for images (`figs/`) and has used `\graphicspath` instead of specifying the subdirectory in `\includegraphics`. This can cause a problem when compiling the main book as the `jmlrbook` class needs to modify the input paths when importing articles to ensure that any relative paths are correctly resolved.

MakeJmlrBookGUI’s new book function solves this problem by removing `\graphicspath` and inserting the subdirectory in the argument of `\includegraphics`.

5. The Doe et al (`DoeJones`) article has multiple `.tex` files. The main file is `main.tex` which includes (via `\input`) the file `results.tex`, which in turn inputs `table1.tex` and `table2.tex`. This won’t in itself cause a problem when the book is compiled, but if the author has made an error in the file, it’s harder to track down which file the problem is in. To overcome this, MakeJmlrBookGUI’s new book function will insert all the included files to create a single `.tex` file.

Note that the new book function creates copies of the source files, rather than modifying the originals, so you will still have the originals for reference or to fall back on if something goes wrong.

The other problem will all these files, is that none of them follow the [PMLR](#) file naming scheme  $\langle lastname \rangle \langle YY \rangle [\langle suffix \rangle]$  where  $\langle lastname \rangle$  is the first author’s

surname,  $\langle YY \rangle$  is the two-digit year of the conference or workshop and  $\langle suffix \rangle$  is a single letter to disambiguate in the event of duplicate names (for example, smith15a and smith15b). The new book function will attempt to name the new files as appropriate. It will also copy over any supplemental files provided they match `*-supp.*` (for example, smith-supp.zip).

To create a new book use the File  $\rightarrow$  New Book... menu item. This will display the dialog shown in Figure 3.

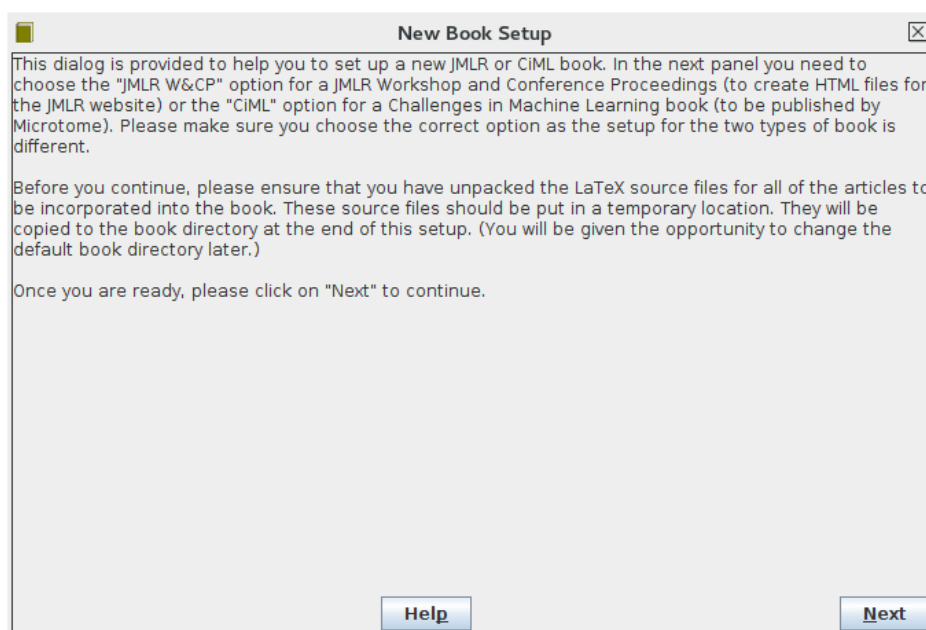


Figure 3: New Book Dialog Initial Screen

When you're ready, click "Next" to move to the next screen, shown in Figure 4. If you are creating a CiML book, click on the "CiML" radio button, which will display the screen shown in Figure 5.

The CiML setting has a button labelled "Import from bib" which you can use to populate the fields from a .bib file previously created for the PMLR according to their specifications. You will need to add the CiML volume number as this isn't stored in the .bib file. The .bib file must be located in the base directory relative to the source files. (In this example, that's the OriginalFiles directory shown above, but the file naming scheme has to match the PMLR requirements. This should be the case if you previously created the proceedings files and generated the .bib file using the Build  $\rightarrow$  Create bib... menu item.)

The volume number is optional for the PMLR setting as it may not have been allocated yet. In this situation, leave the volume field empty and when you do find out the number, change the value of

**New Book Setup**

Type of Book:  JMLR W&CP  CiML

JMLR W&CP Volume:  (Optional)

Title:

Short Title:

Year:

Month:

Day:  (Optional)

Location:  (Optional)

Editor(s):

	Forenames	Surname
Editor 1 Name		

Figure 4: PMLR Title Setup

**New Book Setup**

Type of Book:  JMLR W&CP  CiML

CiML Volume:

JMLR W&CP Volume:  (Optional)

Book Title:

Short Title:  (Optional)

Sub-Title:  (Optional)

Production Editor:  (Optional)



Editor(s):

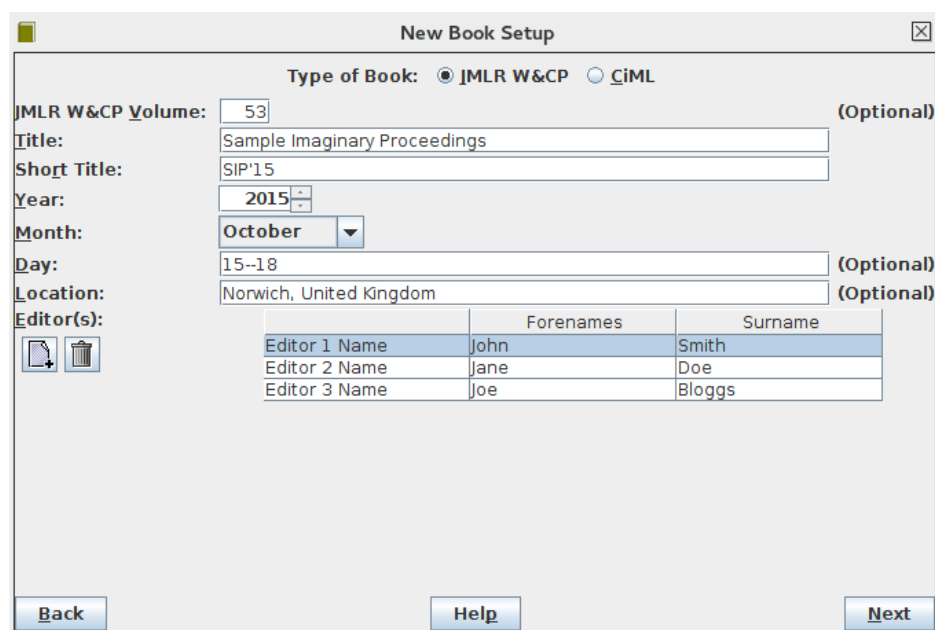
	Forenames	Surname
Editor 1 Name		

Figure 5: CiML Title Setup

```
\jmlrvolume{0}
```

to the appropriate number in the L<sup>A</sup>T<sub>E</sub>X source code.

Each editor is entered in the “Editor(s)” table. The forenames should go in the “Forenames” column and the surname in the “Surname” column. To add a new row, click on the add  button. If you want to delete an editor, select the unwanted row and click on the remove  button. Sample title information for a PMLR volume is shown in Figure 6.



**New Book Setup**

Type of Book:  JMLR W&CP  CiML

JMLR W&CP Volume:  (Optional)

Title:

Short Title:

Year:

Month:

Day:  (Optional)


Location:  (Optional)

Editor(s):

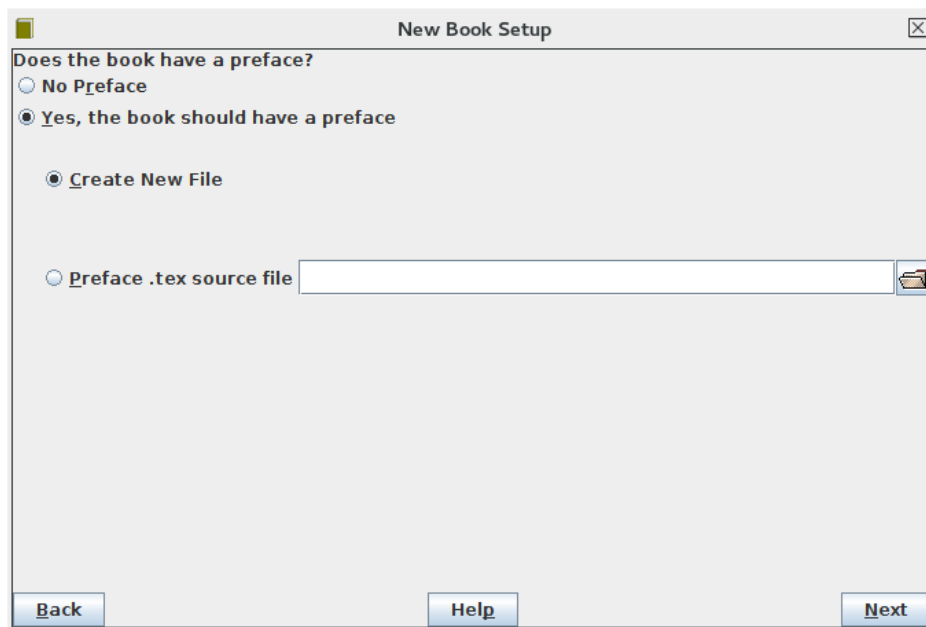
	Forenames	Surname
Editor 1 Name	John	Smith
Editor 2 Name	Jane	Doe
Editor 3 Name	Joe	Bloggs

Figure 6: Sample PMLR Title Information

When you have finished specifying the title information, click on “Next” to move to the next screen. If you have specified a PMLR volume, the screen will be as shown in Figure 7. If you have specified a CiML book, the screen will be as shown in Figure 8.

In both cases, there is the option to have a preface at the start of the book. If the book requires a preface, check the “Yes, the book should have a preface” button. If you already have a file containing the preface text, select the “Preface .tex source file” button and use the file selector to specify the location of this file. (Click on the browse  button to open the file selector.) If the preface hasn’t been written yet, select the “Create New File” button to create a new file.

If you have specified a CiML book, you can do likewise for the foreword. When you are ready, click on “Next” to move to the next screen, shown in Fig-

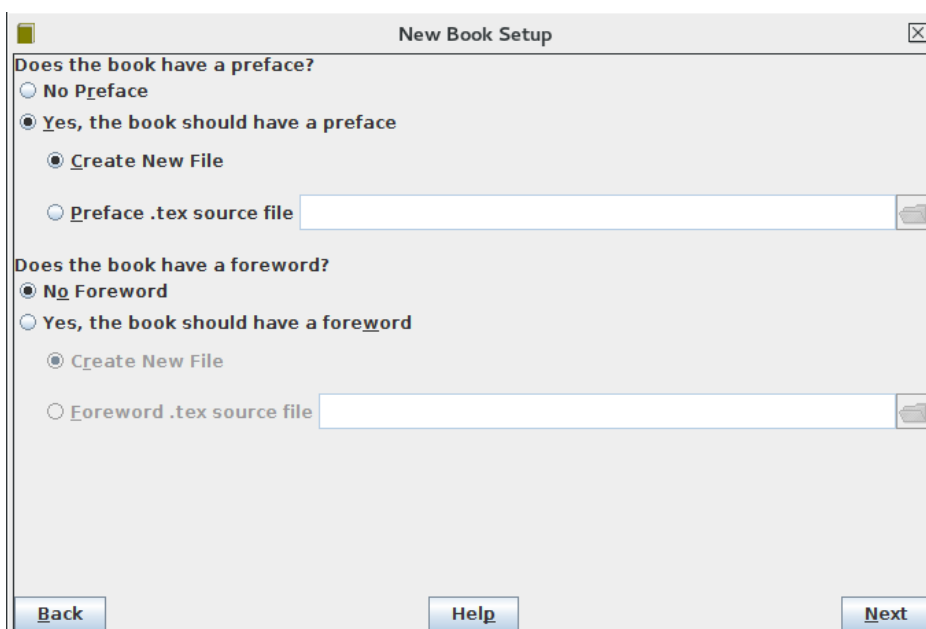


The screenshot shows a window titled "New Book Setup" with a close button in the top right corner. The main content area contains the following text and controls:

- Question: "Does the book have a preface?"
- Radio button:  No Preface
- Radio button:  Yes, the book should have a preface
  - Radio button:  Create New File
  - Radio button:  Preface .tex source file [text input field with file icon]

At the bottom of the window, there are three buttons: "Back", "Help", and "Next".

Figure 7: PMLR Front Matter Setup



The screenshot shows a window titled "New Book Setup" with a close button in the top right corner. The main content area contains the following text and controls:

- Question: "Does the book have a preface?"
- Radio button:  No Preface
- Radio button:  Yes, the book should have a preface
  - Radio button:  Create New File
  - Radio button:  Preface .tex source file [text input field with file icon]
- Question: "Does the book have a foreword?"
- Radio button:  No Foreword
- Radio button:  Yes, the book should have a foreword
  - Radio button:  Create New File
  - Radio button:  Foreword .tex source file [text input field with file icon]

At the bottom of the window, there are three buttons: "Back", "Help", and "Next".

Figure 8: CiML Front Matter Setup

Figure 9. All options are available for [CiML](#) books. None of these options are available for [PMLR](#) volumes.

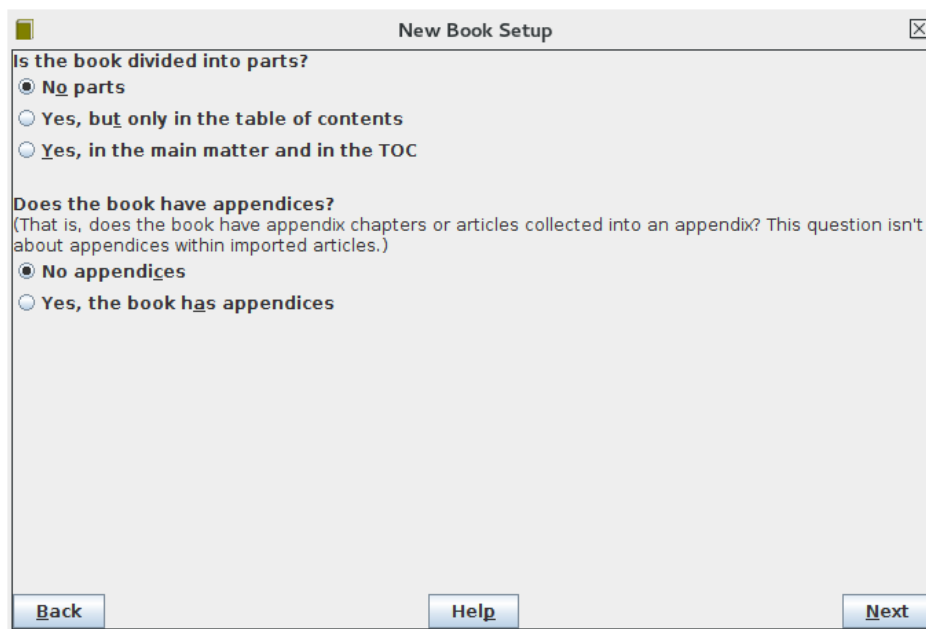



Figure 9: Book Parts and Appendices (CiML Only)

For the [CiML](#) book, if the articles should be grouped into parts (for example, “Part 1: Fundamentals and Theory” and “Part 2: Applications”), then select “Yes, but only in the table of contents” or “Yes, in the main matter and in the TOC”. If the book should also have articles included as an appendix, select “Yes, the book has appendices” otherwise select “No appendices”.

When you are ready, click on “Next” to move to the next screen. This will appear as shown in [Figure 10](#) for [PMLR](#) volumes.

For [CiML](#) books, the screen will vary according to the options from the previous screen. If you have specified no parts and no appendices, the screen will be as shown in [Figure 11](#). If you have specified parts, then the screen will be as shown in [Figure 12](#). If you have specified appendices, there will also be an appendix panel. For example, [Figure 13](#) is for a [CiML](#) book with parts and appendices.

For [CiML](#) books, each import file panel has a check box, which should be selected if the article was originally published in a [PMLR](#) (or [JMLR W&CP](#)) volume. If this box is checked, you need to enter the start and end page numbers from the original [PMLR](#) article.

In the import file panel, use the file selector  to select the article’s `.tex` file. If the author has supplied multiple `.tex` files, you need to make sure you only select the file containing the `\documentclass` command. For example,



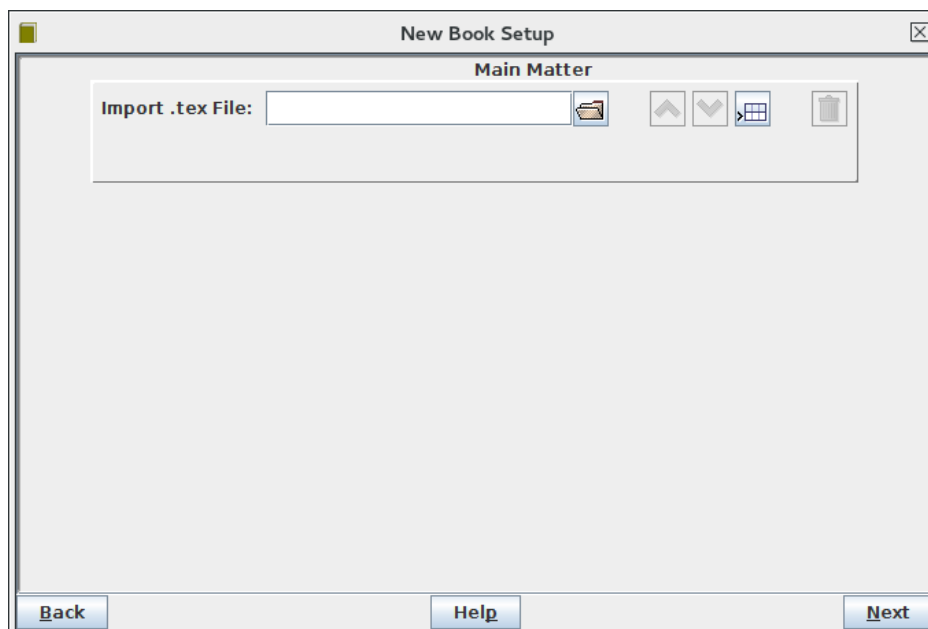


Figure 10: PMLR Import Articles

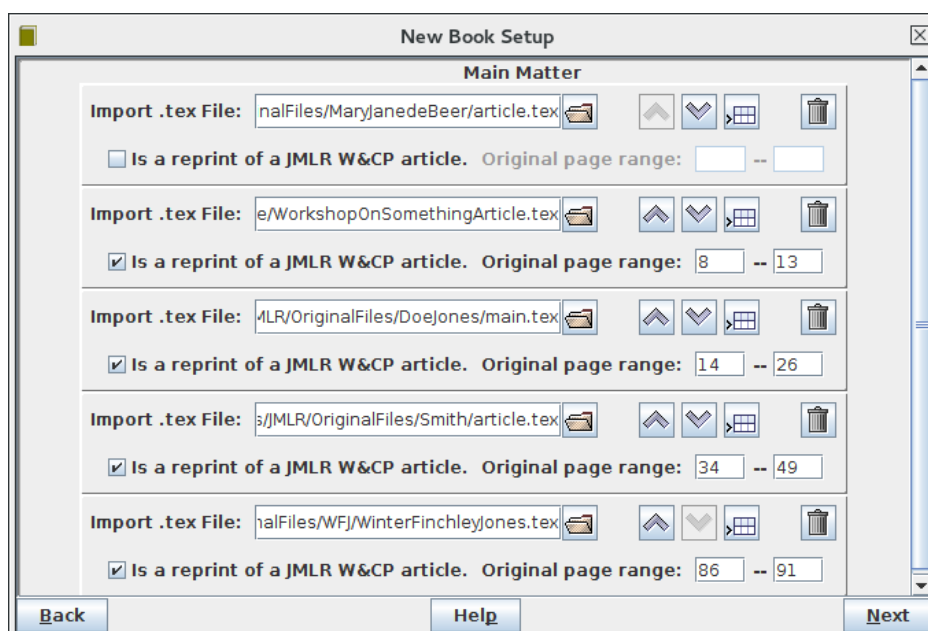


Figure 11: CiML Import Articles (No Parts)

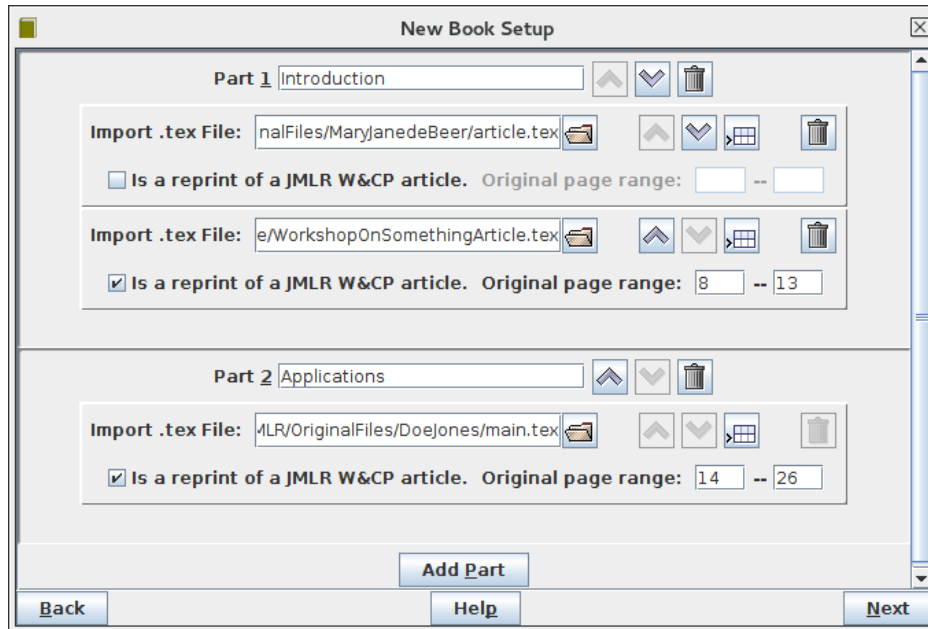


Figure 12: CiML Import Articles (Parts)

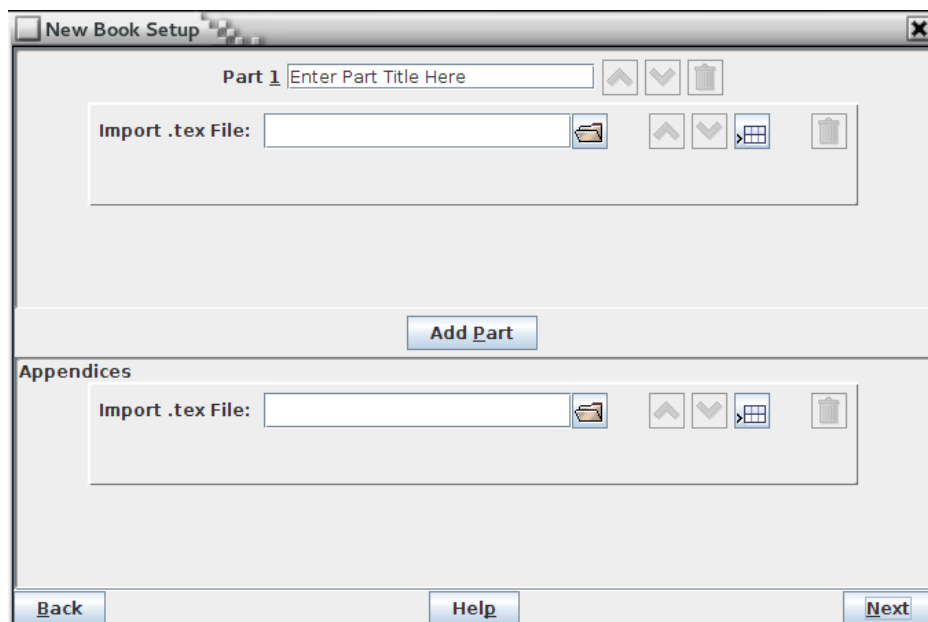


Figure 13: CiML Import Articles (Parts and Appendices)

in [Figure 14](#), I've selected the `article.tex` file. This file isn't a reprint (perhaps it's an introductory article for the [CiML](#) book) so I've left the reprint box unchecked. This reprint option isn't available for the [PMLR](#) version ([Figure 15](#)).

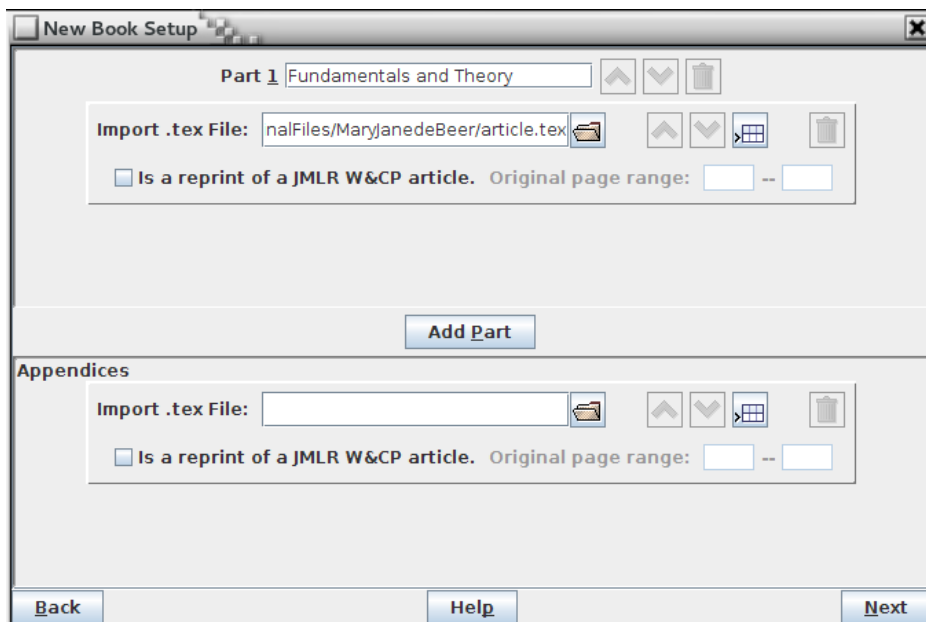





Figure 14: CiML Import Article Specified

To add a new article to the list, click the new row  button to add a new import panel. To add a new part panel, click on the “Add Part” button. You can reorder articles within a part using the up or down arrow buttons. You can remove an import panel by clicking on the associated  button. Drag and drop is supported for this panel, so you can drag import panels up or down to a new location. You can also drag and drop files from your filing system onto the import panels, if your operating system supports this.

In [Figure 16](#) I have selected all my sample files. The first part has the title “Fundamentals and Theory” and has the de Beer and d’Arçé articles. The second part has the title “Applications” and has the Doe and Smith articles. The final article, de Winter et al, is in the appendices section.

When you’re ready, click on the “Next” button to move to the next screen, as shown in [Figure 17](#). This screen specifies in which directory the new book will be created. The default name for this directory is given by “Volume” followed by the volume number. Earlier in this example, I entered “53” for the volume, so the default directory name is `Volume53`. The path this directory resides in is specified in the right-hand field. You can change it using the file selector .

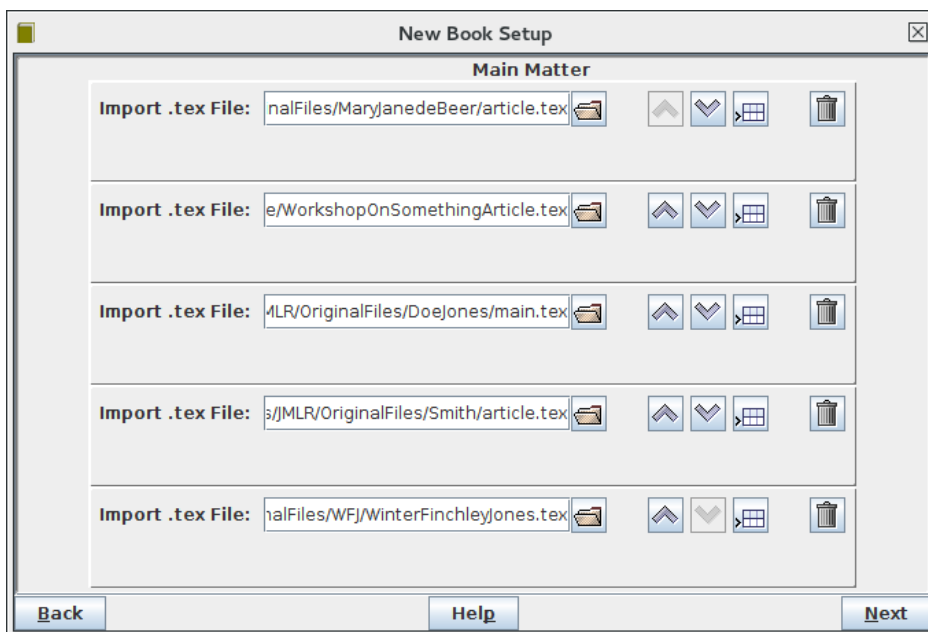


Figure 15: PMLR Import Article Specified

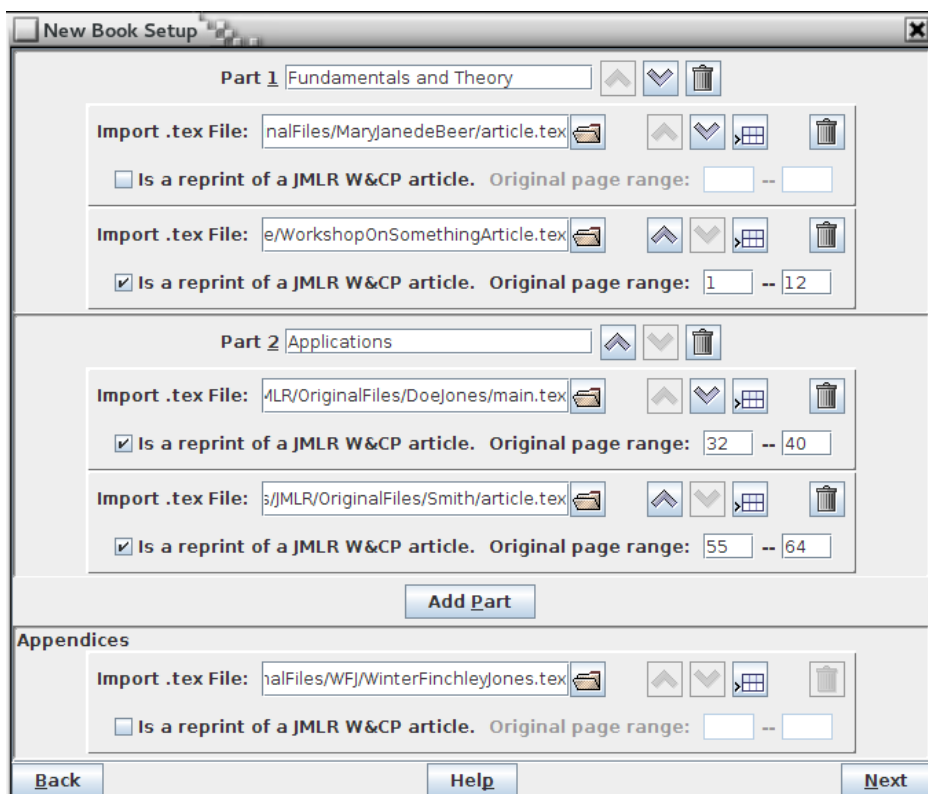


Figure 16: CiML All Import Articles Specified

Note that the book directory can't already exist. This is a precautionary measure to ensure you don't accidentally overwrite the original files, in case they are still needed.

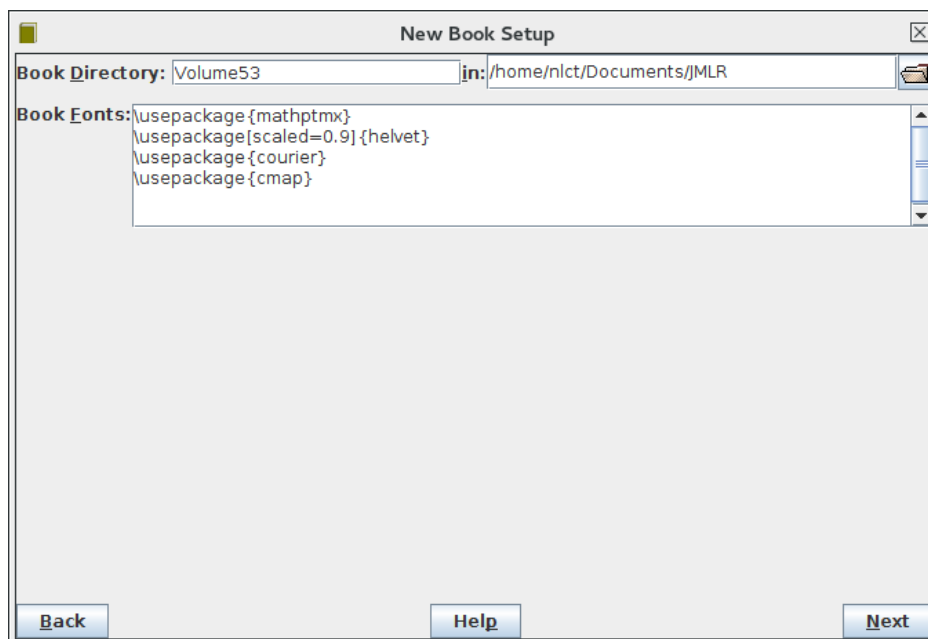


Figure 17: New Book Directory

When you are ready, click on the “Next” button. This will start the new book creation process. Recall from earlier that the Winter et al example had a custom package `newstuff.sty` provided by the author. Since this is likely to cause a problem, an error message is shown (Figure 18). This identifies the problem file and the line number where the `\usepackage` command occurred. There's a possibility that the package may actually be available on [the Comprehensive TeX Archive Network \(CTAN\)](#) but isn't installed on your computer. In which case you're provided with a URL that you can copy and paste into your web browser to check.<sup>2</sup> If it turns out that the package is available on [CTAN](#), you will need to install it and then edit the article's new `.tex` file to uncomment the relevant `\usepackage` command, and you will also need to edit the book's `.tex` file to add it to the list of required packages.

Once you dismiss this message box, you can view the transcript in the final screen of the new book dialog (see Figure 19). If you want a copy of this transcript, you can copy it to the clipboard using the “Copy” button. If you want to view the newly created book file, click on the “View main `.tex`” button.

<sup>2</sup>The fact that `mystuff.sty` is present in the article's directory doesn't mean that it's the author's file as authors often copy all the packages they use into their article's directory.

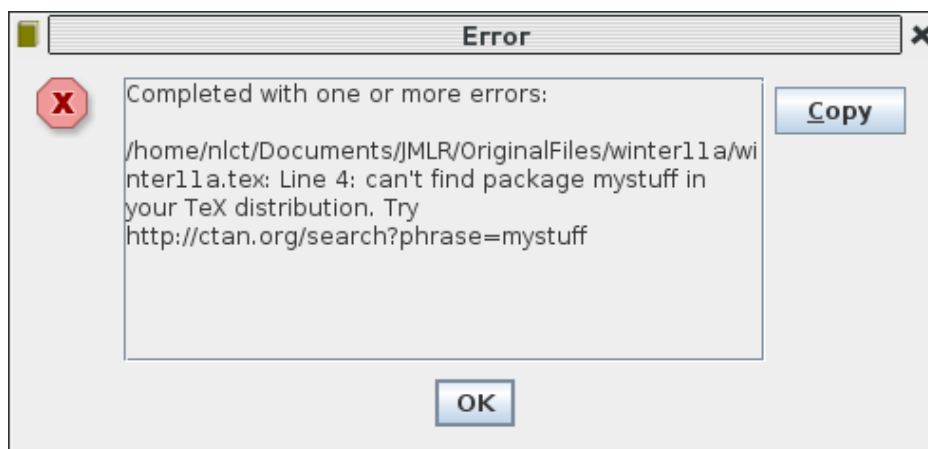


Figure 18: Error Message

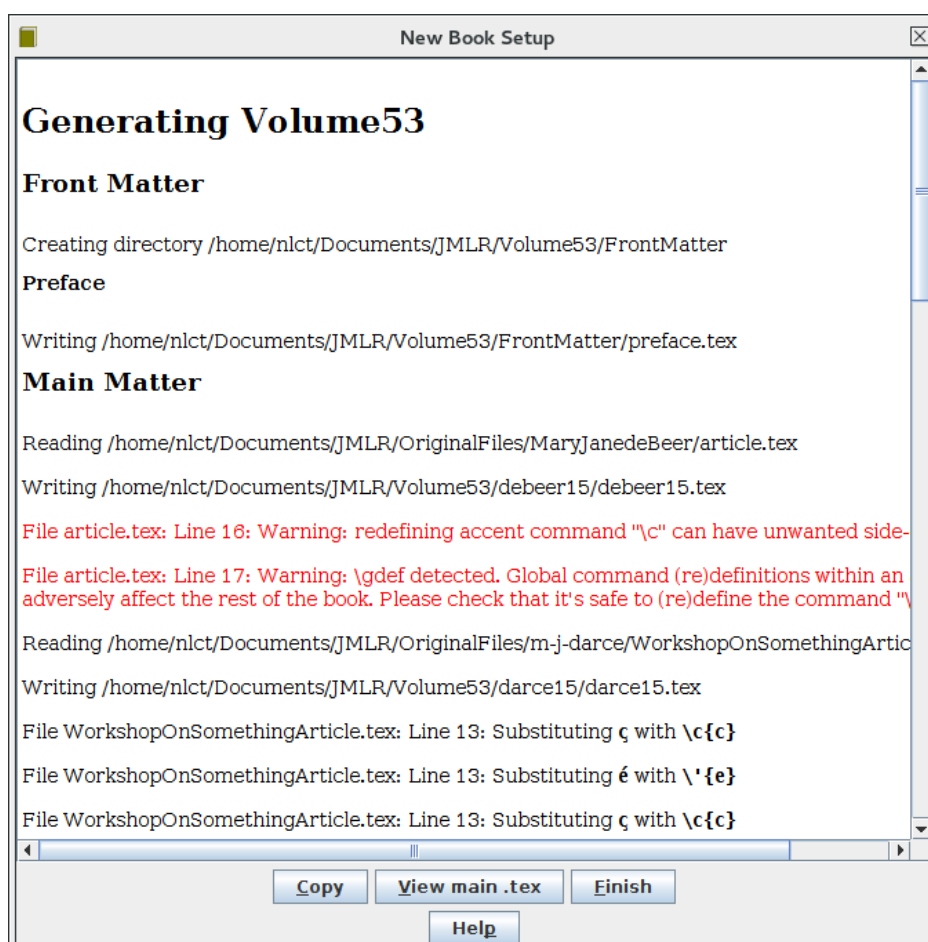


Figure 19: Transcript

You can now either close the new book dialog by clicking on the close window button in the title bar, which will do nothing further, or you can click on the “Finish” button, which will close the dialog and load the new book into MakeJmlrBookGUI.

## 5 Loading a Book

Use the File → Open Book... menu item to specify the main file for the PMLR volume you want to process (Figure 20). Alternatively, you can use the File → Recent Book menu to select a recently loaded file. Note that the main file is the file that loads the jmlrbook class using:

```
\documentclass[pmlr]{jmlrbook}
```

If you don’t yet have a main file, you can create one with the new book function (see §4 Creating a New Book).

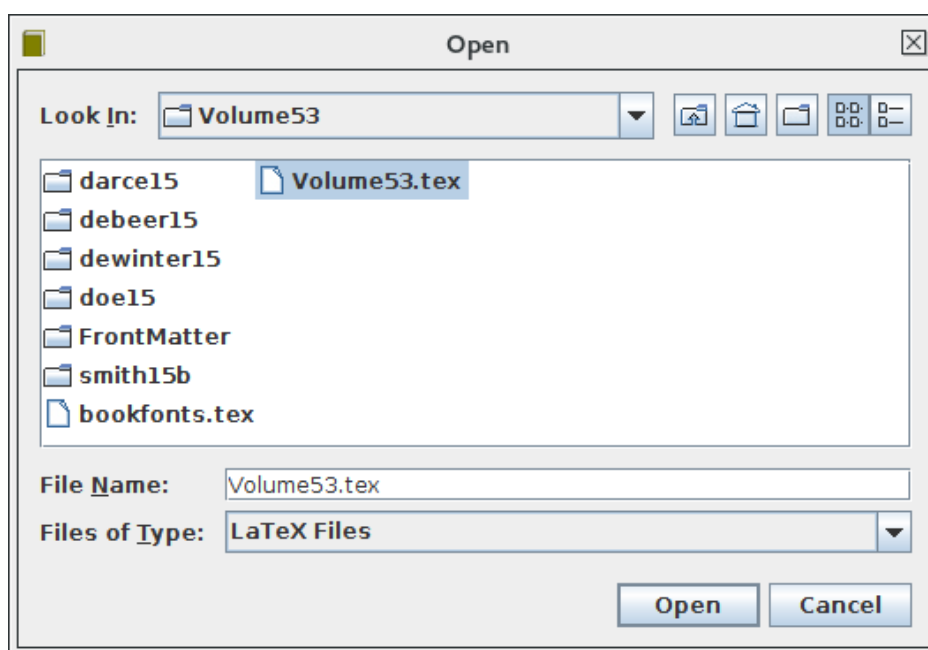


Figure 20: Opening a PMLR Volume

Once a book has been loaded  $\text{\LaTeX}$  (and possibly  $\text{\BibTeX}$ ) will automatically be run to create an up-to-date version of the book.

To clear the book details so that you can open a new one, use the File → Clear menu item.

## 6 The Build Menu

Different formats of the book can be created using the Build menu, which becomes enabled once you have [loaded](#) a book.

**Discrepancies in the page numbering can occur if the book uses a different set of fonts to the included articles.**

If you want the book to use a particular set of fonts, create a file called `bookfonts.tex` with the commands to load the font packages and add it the book's main directory. `MakeJmlrBookGUI` will check the main tex file for the occurrence of

```
\input{bookfonts}
```

and if found will input that file when building the individual papers. (Note that when building the articles this inputs `../bookfonts`, so the articles must be in immediate subdirectories of the main book directory.)

### 6.1 Building the Main Book PDF

The entire book can be created using the Build → Book (Color Hyperlinked PDF) menu item. Since the `jmlr` and `jmlrbook` classes use the `hyperref` package, the resulting PDF will have hyperlinks and a set of PDF bookmarks. You may find the `bookmark` package better at forming the bookmarks, but that package must be loaded last. Note that although you need to build the entire book in order for `MakeJmlrBookGUI` to build all the standalone articles with the correct page numbering, you shouldn't send the entire book PDF to the [PMLR](#) production editor. (Instead, just send the `.zip` file containing the articles and `.bib` file as described in [§6.5 Create Proceedings zip File](#).)

### 6.2 Building the Print Version

If the book is to be printed, you need to supply the publisher with a PDF/1-X compliant document. `MakeJmlrBookGUI` tries to make a grey scale compliant document using `ghostscript`. However it is recommended that you check that the resulting document is compliant. This option is just for [CiML](#) books, not for [PMLR](#) volumes. Avoid transparency with this option.

### 6.3 Building Individual Papers

It may be that you are working on one particular paper and don't want to have to keep rebuilding the entire book to check your changes. In which case, you can use the Build → Papers sub-menu.



## 6.4 Create Proceedings bib File

You can create the `.bib` file required for the [PMLR](#) volumes with the Build → Create bib... menu item.

This function isn't required for [CiML](#) books.

## 6.5 Create Proceedings zip File

You can create the `.zip` file with all the files for submission to the [PMLR](#) production editor with the Build → Create zip... menu item. If the `.bib` file hasn't already been created it will be created by this function and added to the `.zip` archive. All supplemental files must have the naming scheme as described in the [PMLR Proceedings Specifications](#). All matching files will automatically be added to the `.zip` file.

This function isn't required for [CiML](#) books.

# 7 The Tools Menu

Use Tools → Remove Aux to remove all the auxiliary files. If you want to abort a current process, use Tools → Abort.

## 7.1 Check Characters Outside the Range 0x20–0x7E

Encoding problems can arise when different authors have used different input encodings. For example, one article may use latin1 and another may use utf8. Even more troublesome is where a single file contains a mixture of encodings. This can happen if the co-authors haven't agreed on an encoding, or (quite often) entries in the bib file have been copied and pasted from different sources so that different bibliography entries use different encodings. Sometimes you will get a message about missing or undefined characters, but sometimes (as in [Example 6](#) in [§9 Troubleshooting](#)) there will be no messages but the resulting PDF file will appear incorrect.

To help editors track down problem characters, `MakeJmlrBookGUI` version 0.4.2b has introduced the Tools → Character Check... menu item. Use this function to open a dialog box with a list of all the `.tex` and `.bib` files belonging to the book. The files are arranged in hierarchical order. If you select a directory, everything inside that directory will also be selected, so to select all files just click on the root node. Once you have selected all the files you want to check, click "Okay" and the application will check each file for characters outside the range 0x20 (space) to 0x7E (tilde). On completion, a window will appear with the list

of files with the line numbers where characters outside that range were detected. (The characters will be highlighted in bold red to make them stand out.)

## 8 Application Settings

Application settings can be modified using the Settings → Edit... menu item, which will open the Properties dialog box. This has five tabs:

### Book

If the “Conform to publisher requirements” checkbox is selected, the application will run checks for certain requirements, such as checking that file names follow the PMLR convention. Note that not all PMLR requirements are checked and some have been changed since this code was written.

### Startup Directory

You can specify which directory MakeJmlrBookGUI should start up in. This can be the current working directory, the last directory used by the application or a specific directory.

### Applications

You need to set the paths to executables required by the [build processes](#). They are:

**Path to LaTeX** The path to the PDF $\LaTeX$  executable is required. Note that the `jmlr` and `jmlrbook` classes were optimised to use PDF $\LaTeX$ . You are likely to run into errors if you use  $\LaTeX$  with DVI output.

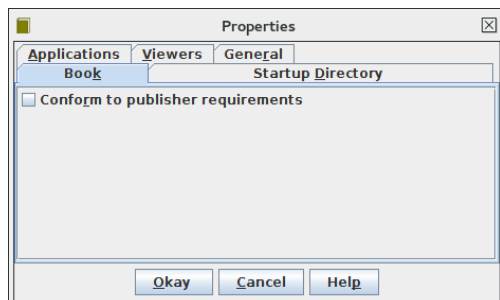
**Path to BibTeX** The path to the BibTeX executable is required to compile bibliographies supplied with BibTeX databases.

**Path to GhostScript** The path to `ghostscript` is required when building the print version of the book.

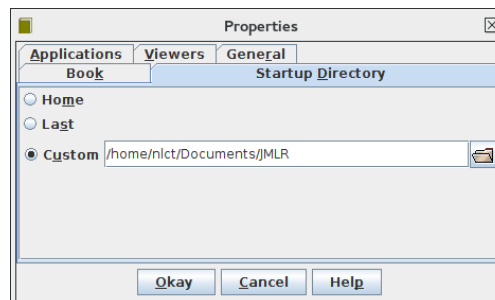
**Path to kpsewhich** The path to `kpsewhich` is required to located files in your  $\TeX$  distribution.

### Viewers

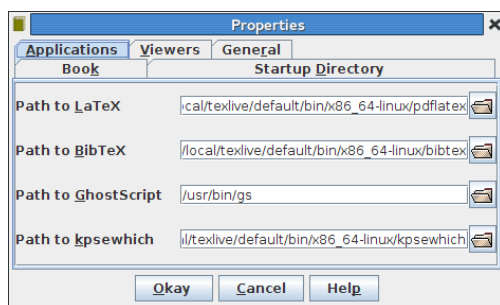
You need to set paths in order to view or edit files when following links in the book tab or by double-clicking on files in the [Book File List Tab](#).



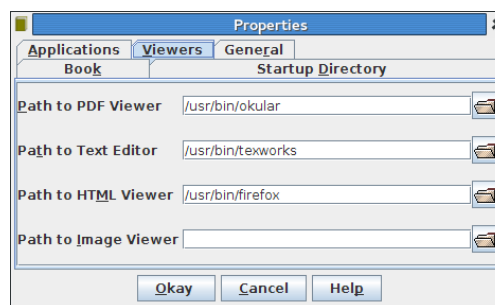
(a)



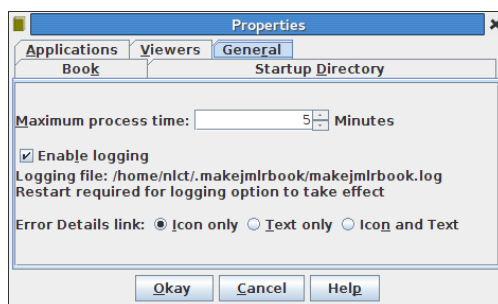
(b)



(c)



(d)



(e)

Figure 21: Properties Dialog: (a) book settings; (b) startup settings; (c) paths to build applications; (d) paths to file viewers and text editor; (e) general settings.

**Path to PDF Viewer** The path to a PDF viewer is required if you want to click on any of the PDF links in the main panel.

**Path to Text Editor** The path to a text editor is required if you want to click on any of the links to source code or log files. The text editor may be a general purpose text editor, such as `gedit`, or it can be a  $\text{T}_{\text{E}}\text{X}/\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  front end, such as TeXWorks. Since TeXWorks is cross-platform, MakeJmlrBookGUI will default to that if the TeXWorks executable is found on the system path.


**Path to HTML Viewer** The path to an HTML viewer is required for viewing HTML files.

**Path to Image Viewer** The path to an image viewer is only required if you want to load an image from the file list tab.

## General

To help prevent orphaned processes, each spawned process (such as a  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  run) is given a maximum process time. The default is 5 minutes, which is usually sufficient for a normal  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  run, but more time may be required when converting a PDF file to PDF/X, as is done by the build print version function. Large books with a lot of included images can take considerably longer, in which case you will need to change the maximum process time *before you run the process*. Changing this value won't affect currently running processes.

For debugging purposes, you can also switch on the logging file in this tab. You must quit and restart the application to enable logging. The log file is overwritten at the start of the application. Make sure you first quit the application before viewing the log file to ensure that it is closed safely.

Some of the error messages display in the main panel have a link next to them that opens a dialog window with further details and recommendations. The default link shows the information icon . On some operating systems or Java versions, this icon doesn't seem to show. If this is the case, or if you prefer text rather than an image, you can select the appropriate "Error Details link" radio button.

## 9 Troubleshooting

MakeJmlrBookGUI tries to detect common problems encountered when using the `jmlrbook` class. The navigation panel shows the problem papers in red to easily

identify them. As mentioned at the end of the previous section, some of these error messages have a link next to them to provide further details or recommendations.

Try to fix the errors in the individual papers before you try addressing the errors for the entire book. Most of the errors listed here shouldn't occur if you set up the book using the File → New Book... function.

**Be careful of imported articles that use a different input encoding to the main book! This will not always produce an error message, but characters will not appear correctly in the PDF file. (See Example 6.)**

## 9.1 Examples of common problems

(The images here are from an older version with a slightly different interface.)

1. [Figure 22](#): one of the imported papers (beer11a) has an error in it. Clicking on the red “beer11a” link in the navigation panel scrolls the main panel to the relevant entry. In this case the error is an undefined control sequence which has been used on line 31. It may be that the command was defined in an accompanying package that hasn't been loaded.
2. [Figure 23](#): one of the imported papers (smith11a) has a copy of an old version of the jmlr class in its directory. This unwanted file should be deleted.
3. [Figure 24](#): one of the papers (paper1) in this book has used a package (theorem) that is known to conflict with the jmlr and jmlrbook classes. The imported document must be modified so that it no longer uses that package. This is an issue that is corrected by the new book function.

Note that this example has also produced non-conformity error messages (shown in purple).

4. [Figure 25](#): one of the papers in this book (paper3) is not using the jmlr class. (It's using the article class.) This is the reason for all the obscure error messages. It's also using obsolete packages (jmlr2e and epsfig). These problems must all be fixed. This is an issue that is corrected by the new book function.
5. [Figure 26](#): one of the papers was misspelt in the main book file. Instead of:

```
\importpaper{smith11a}{smith11a}
```

it was written as

```
\importpaper{smith11a}{smoth11a}
```

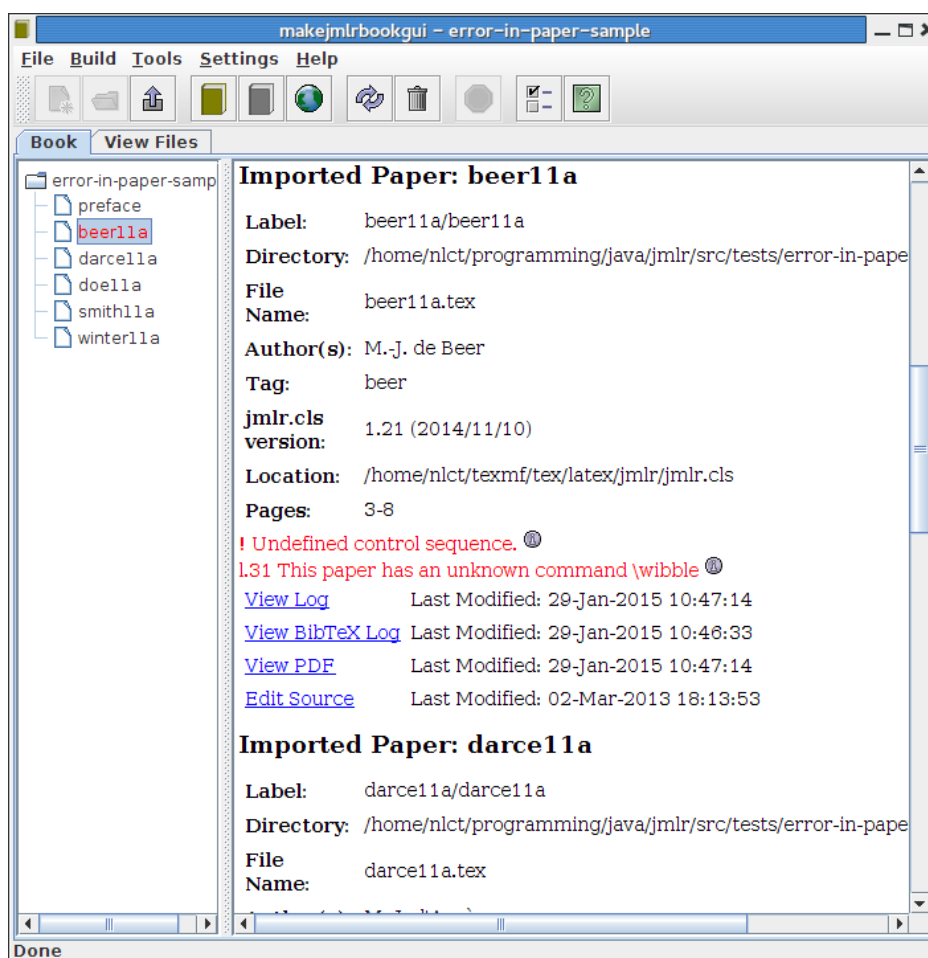


Figure 22: One of the imported papers (beer11a) has an error in it.

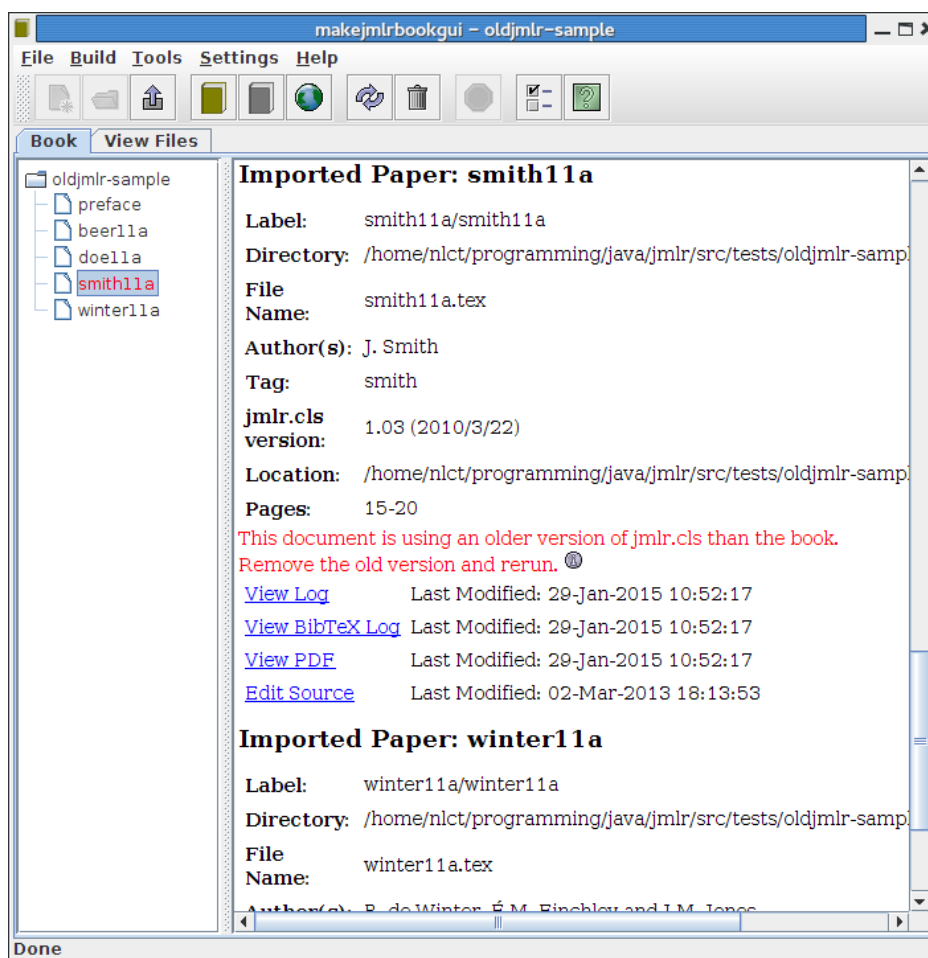


Figure 23: One of the imported papers (smith11a) is using an different version of jmlr.cls from the rest of the book.

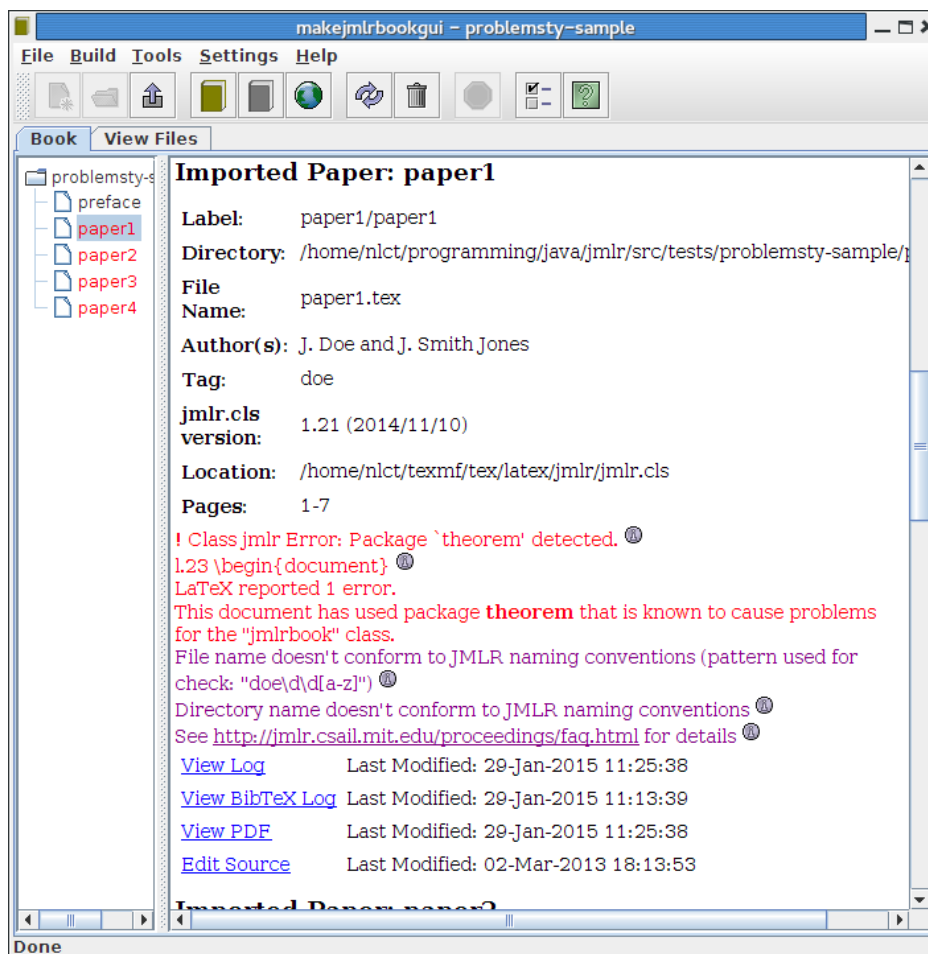


Figure 24: One of the imported papers is using a package that is known to conflict with the jmlr bundle.



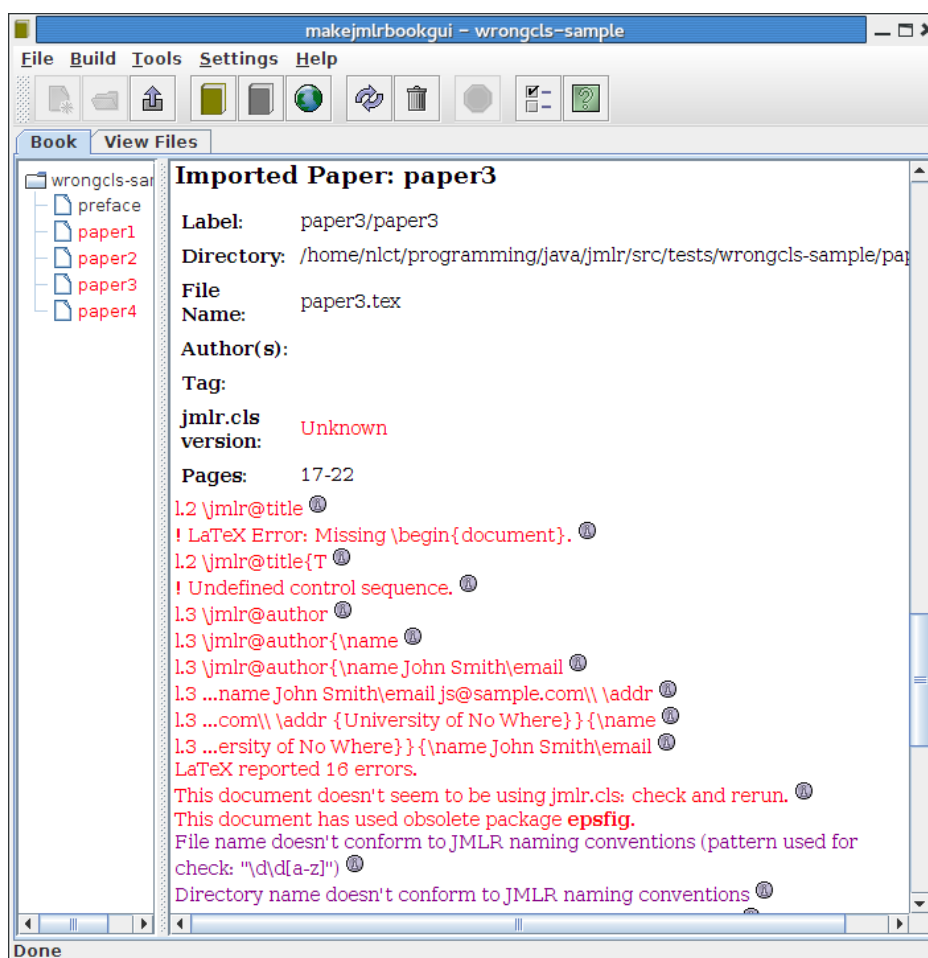


Figure 25: One of the imported papers is not using the jmlr class.

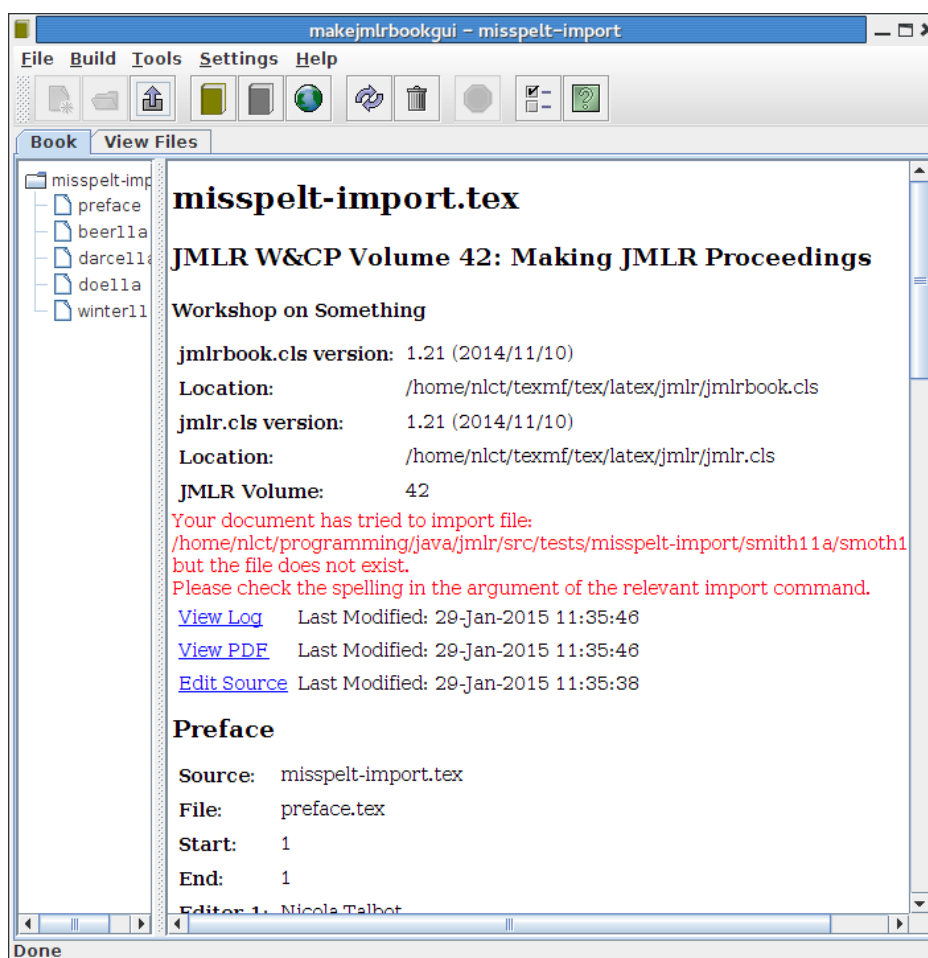


Figure 26: One of the imported papers has been misspelt in the main book file.

This error shouldn't occur with the new book function.

6. Imported paper `darcella/darcella.tex` has used `utf8` encoding via:

```
\usepackage[utf8]{inputenc}
```

but the main book document has used `latin1` encoding via:

```
\usepackage[latin1]{inputenc}
```

The imported article has specified the author name using a Unicode diacritic:

```
\title[D'Arcé Article]{An Article by Mary-Jane d'Arcé}
```

```
\author{\Name{Mary-Jane {d'Arcé}}\Email{mjdb@sample.com}\
\addr{University of No Where}}
```

This appears correctly for the standalone article, but in the book the surname appears as d'Arcé© instead of d'Arcé. Imported articles that use the `inputenc` package can switch the encoding using `\inputencoding`. In this example, `darcella/darcella.tex` could have:

```
\usepackage[utf8]{inputenc}
\inputencoding{utf8}
```

however this might corrupt the book's table of contents if the non-ASCII characters appear in the headings or author names (as in this case). See [§7.1 Check Characters Outside the Range 0x20–0x7E](#) to help detect this type of problem.

The new book function tries to detect characters that have a known  $\LaTeX$  equivalent and replace them accordingly, but it doesn't check the `.bib` files which may contain non-ASCII characters.

7. [Figure 27](#): one of the papers (`winter11a`) has loaded `siunitx`, but the main book file hasn't. To correct the problem, add the package to the main book file. Remember that all packages that must be loaded before `hyperref` should be included as follows:

```

\def\jmlrprehyperref{
  \usepackage{siunitx}
  \usepackage{booktabs}
}
\documentclass[pmlr]{jmlrbook}

```

This is something that the new book function does automatically, provided the package is installed in your  $\text{\TeX}$  distribution.

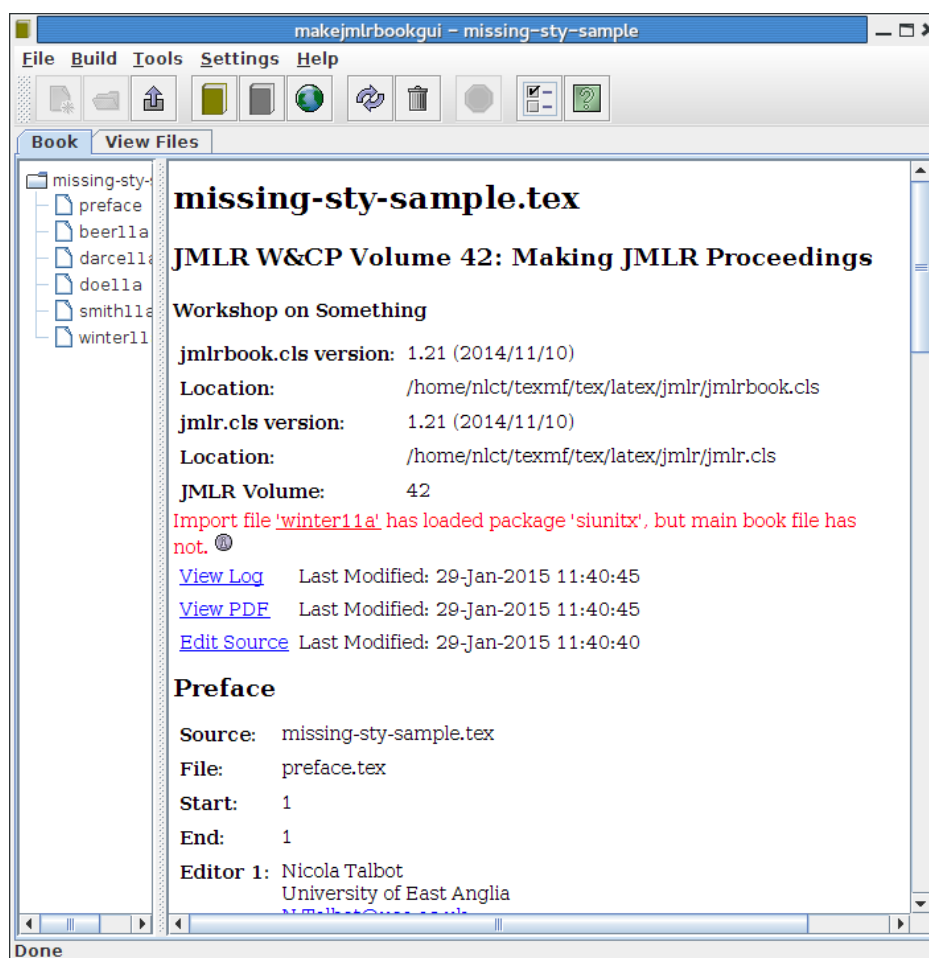


Figure 27: One of the imported papers (winter11a) has loaded a package (siunitx) that has not been loaded by the main book file.

8. [Figure 28](#): one of the imported papers (doe11a) contains a copy of an old version of the xfrac package. This old copy should be removed so that all files use the same version of that package.

The new book function doesn't copy across any packages that the authors may provided with their document source, but instead checks that the packages required are installed in your TeX distribution.

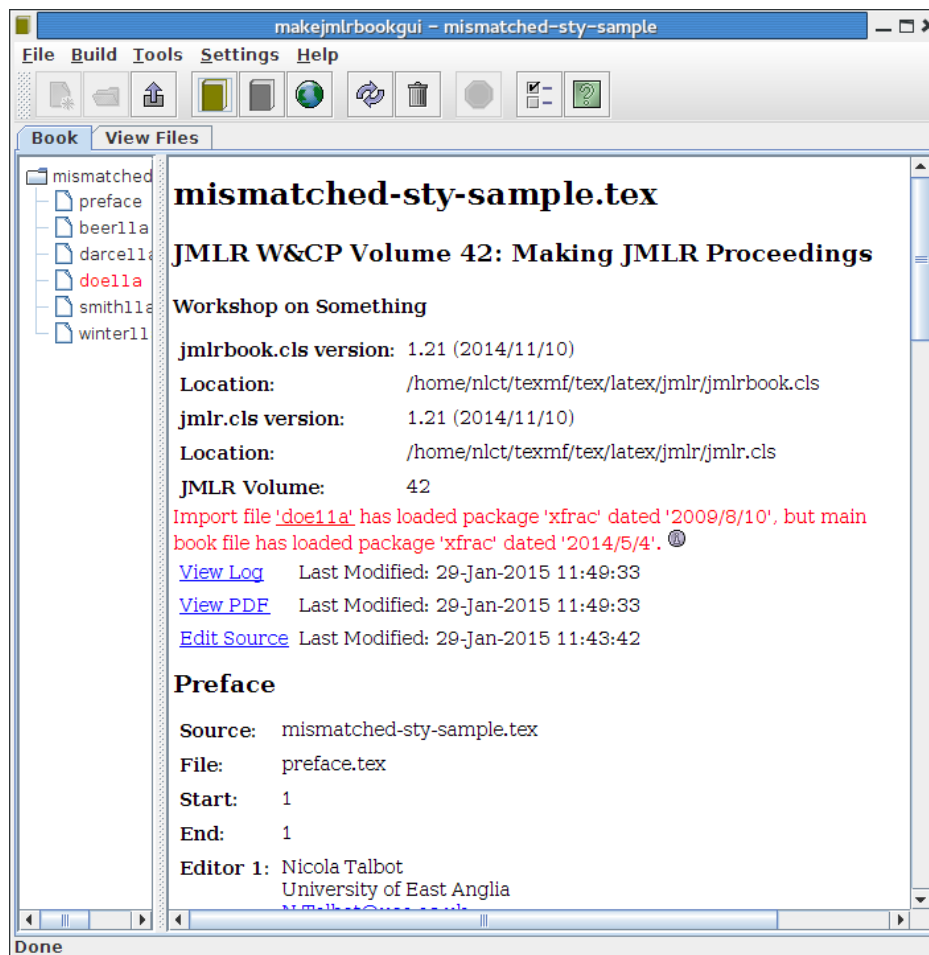


Figure 28: One of the imported papers (doe11a) has loaded a different version of a package loaded by the main book file

9. Suppose you have included articles in your book using `\importpaper` and all the paper ranges in the book pdf seem correct, but when you look at the individual article PDFs there is a discrepancy in the page range. Why is this?

Most likely, it's because the article is using a different set of fonts to the book. Perhaps the article is using the default Computer Modern, but the book is using Times, Courier and Helvetica. This change in font size can occasionally be enough to shift the last page. Since it's a bit inconvenient to

have to edit every imported article to use the same set of fonts as the book, MakeJmlrBookGUI checks the main book .tex file for the line

```
\input{bookfonts}
```

If found, it will try to include this file when building the standalone articles. (See §6 The Build Menu.)

The new book function creates a file called `bookfonts.tex` with the font packages specified in the setup dialog.

## 9.2 Error Messages

### 1. Message:

```
Exception in thread "main"  
java.lang.UnsupportedClassVersionError:
```

Cause: you are using a version of Java that is too old. You need Java 7, which can be downloaded from <http://www.oracle.com/technetwork/java/javase/downloads/>.

### 2. Message:

```
Process timed-out (process time limit: 300000ms)
```

Cause: a spawned process has been aborted because it has taken too long to complete. The default time-out value is 300000ms (5 minutes). This most-likely means that the process has got stuck, but if you have a very slow machine and a very large book, you may need to increase the time-out threshold using the [-timeout command line switch](#).

### 3. LaTeX Message:

```
! LaTeX Error: Command \c@lenddoca already  
defined. Or name \end... illegal, see p.192 of  
the manual.
```

Cause: conflict between `combine.cls` and `hyperref.sty`. You need to upgrade to at least version 1.16 of `jmlr.cls` and `jmlrbook.cls`.

## **Abbreviations**

### **CiML**

Challenges in Machine Learning. [1](#), [6](#), [10](#), [12](#), [14](#), [17](#), [22](#), [23](#)

### **CTAN**

the Comprehensive TeX Archive Network. [19](#)

### **JMLR W&CP**

Journal of Machine Learning Research Workshop and Conference Proceedings. [1](#), [14](#)

### **PMLR**

Proceedings of Machine Learning Research. [1](#), [6](#), [9](#), [10](#), [12](#), [14](#), [17](#), [21–24](#)

## Index

- `\appendix`, 3
- article, 8, 27
- bookmark, 22
- Build menu, 22
  - Book (Color Hyperlinked PDF), 22
  - Create bib..., 10, 23
  - Create zip..., 23
  - Papers, 22
- `\chapter`, 3
- combine, 2
- `\documentclass`, 2, 14
- `\epsfig`, 8
- epsfig, 8, 27
- File menu
  - Clear, 21
  - Export Messages..., 6
  - New Book..., 10, 27
  - Open Book..., 21
  - Recent Book, 21
- gedit, 26
- ghostscript, 22, 24
- `\graphicspath`, 9
- hyperref, 22, 33
- `\importpaper`, 35
- `\includegraphics`, 8, 9
- `\input`, 8, 9
- inputenc, 33
- `\inputencoding`, 33
- jmlr, 1, 1, 4, 8, 22, 24, 27
- jmlr2e, 8, 27
- jmlrbook, 1, 1, 2, 8, 9, 21, 22, 24, 26, 27
- jmlrpapers, 3
- kpsewhich, 24
- latex, 8
- makejmlrbookgui, 2
- mktexlsr, 5
- `\newcommand`, 9
- pdflatex, 8
- `\renewcommand`, 9
- Settings menu
  - Edit..., 24
- siunitx, 33
- texhash, 5
- theorem, 27
- Tools menu
  - Abort, 23
  - Character Check..., 23
  - Fetch JMLR LaTeX Bundle, 4
  - Refresh Files..., 6
  - Remove Aux, 23
- `\usepackage`, 19
- xfrac, 34



## A Licence

# GNU GENERAL PUBLIC LICENSE VERSION 3, 29 JUNE 2007

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

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

## PREAMBLE

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

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

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

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

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

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

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

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

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

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

## TERMS AND CONDITIONS

### 0. Definitions.

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

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

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

To “modify” a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact

copy. The resulting work is called a “modified version” of the earlier work or a work “based on” the earlier work.

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

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

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

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

#### 1. Source Code.

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

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

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

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

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

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

## 2. Basic Permissions.

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

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

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

## 3. Protecting Users’ Legal Rights From Anti-Circumvention Law.

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

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

#### 4. Conveying Verbatim Copies.

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

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

#### 5. Conveying Modified Source Versions.

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

- (a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- (b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- (c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

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

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

#### 6. Conveying Non-Source Forms.

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

- (a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- (b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- (c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- (d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further

charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

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

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

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

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

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of

how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

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

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

## 7. Additional Terms.

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

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

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

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



- (b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- (c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- (d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- (e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- (f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

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

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

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

## 8. Termination.

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

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until

the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

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

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

#### 9. Acceptance Not Required for Having Copies.

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

#### 10. Automatic Licensing of Downstream Recipients.

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

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

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under

this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

#### 11. Patents.

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

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

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

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

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

If, pursuant to or in connection with a single transaction or arrangement,

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

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

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

#### 12. No Surrender of Others’ Freedom.

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

#### 13. Use with the GNU Affero General Public License.

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

the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

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

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

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

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

15. Disclaimer of Warranty.

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

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER

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

17. Interpretation of Sections 15 and 16.

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

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Programs

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

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

```
<one line to give the program's name and a brief idea of what it does.>
```

```
Copyright (C) <textyear> <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/>](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>