

User Manual for MakeGlossariesGUI version 2.2

Nicola L. C. Talbot
www.dickimaw-books.com

2020-03-09

Contents

1 Introduction	3
2 Basic Use (GUI Mode)	5
3 Settings	23
3.1 Start Up Directory	23
3.2 Diagnostic Settings	23
3.3 Indexing Applications	24
3.4 GUI Preferences	25
4 Licence	28
Glossary	44
Index	45

1 Introduction

The \LaTeX glossaries package has three methods of generating a glossary (list of abbreviations, terms or symbols): directly using \LaTeX , using `makeindex` or using `xindy`. The first option requires two \LaTeX runs (as with cross-references), the last two options requires a \LaTeX run followed by running the indexing application (`makeindex` or `xindy`), followed by another \LaTeX run. The `glossaries-extra` package provides two more methods (just using `\printunsrtglossary` or using `\printunsrtglossary` with `bib2gls`). These last two methods aren't supported by `MakeGlossariesGUI` (although it will flag documents that require `bib2gls` if detected, as part of its diagnostics). Your document may require additional applications, such as `bibtex`, but that's outside of the scope of `MakeGlossariesGUI`.

The `glossaries` package provides a Perl script called `makeglossaries` that's intended to simplify this step for options 2 and 3. However, sometimes things can go wrong and users may not be able to work out why. The `makeglossaries` script tries to diagnose problems, but not everyone wants to install Perl for some reason (which is a shame, as there are some useful \TeX -related Perl scripts, including `xindy`). The `glossaries` package also provides a light-weight Lua alternative, `makeglossaries-lite.lua`, but that doesn't provide any diagnostics. `MakeGlossariesGUI` is a Java application that (when run in batch mode) can be used instead of `makeglossaries` or `makeglossaries-lite.lua` however its primary purpose is a **graphical user interface (GUI)** tool for determining what's gone wrong when the glossary doesn't appear or is incomplete.

Since `MakeGlossariesGUI` is written in Java, you'll need the Java Runtime Environment installed. If your document build process supports the conditional execution of commands, then you can test if `MakeGlossariesGUI`, `makeglossaries` or `makeglossaries-lite.lua` are required by searching the log file for the presence of the command `\@istfilename`. (The argument is the style file. The extension determines whether `makeindex` or `xindy` is required.)

The `MakeGlossariesGUI` application runs in **GUI** mode by default, but can be run in batch mode using the `--batch` command line option. Command line invocation:

```
makeglossariesgui [<options>] [<filename>]
```

The *<filename>* must be supplied when run in batch mode (the `.aux` extension may be omitted), but is optional in **GUI** mode. The *<filename>* should be the auxiliary file produced by the \LaTeX run, but may also be the `.tex` or `.log` file if it has the same base-name as the auxiliary file and is located in the same directory. Unlike `makeglossaries` and `makeglossaries-lite.lua`, *<filename>* may include the directory path.

Available options:

- batch (or -b)** Invoke MakeGlossariesGUI in batch mode (*<filename>* must be supplied).
Note that the extra checks that parse the log file aren't performed in batch mode.
- gui** Invoke MakeGlossariesGUI in **GUI** mode (default).
- quiet** Suppress (non-error) messages that would otherwise have been written to STDOUT.
- dry-run (or -n)** Dry run mode (don't run the indexing application).
- nodry-run** Not dry run mode (default).
- debug** Print debug messages to STDOUT.
- version (or -v)** Print the version details to STDOUT and exit.
- help (or -h)** Print a brief summary of available options to STDOUT and exit.

When run in batch mode, MakeGlossariesGUI behaves much like **makeglossaries**. It reads the auxiliary file to determine whether to use **makeindex** or **xindy** and what options to pass to them. The **GUI settings** will be honoured.

2 Basic Use (GUI Mode)

A file can be loaded from the command line invocation (see [chapter 1](#)) or in the GUI using the File→Open menu item. You can also use the load file button on the toolbar or (if permitted by your operating system) drag and drop the file onto the main window. The input file should be the `.aux` file created by L^AT_EX, but if you try to load the main document `.tex` file or the `.log` transcript file, MakeGlossariesGUI will assume you meant the associated `.aux` file. (If you have used L^AT_EX's `-jobname` or `-output-directory` options, then you won't be able to use the `.tex` file and will have to use either the `.aux` or `.log` file.)

Example 1.

Suppose you have the following document (called, say, `basic-sample.tex`):

```
\documentclass{article}

\usepackage{glossaries}

\makeglossaries

\newglossaryentry{sample}{name={sample},
  description={an example}}

\begin{document}
A \gls{sample} document.

\printglossaries

\end{document}
```

First run L^AT_EX the usual way. This should create the auxiliary file `basic-sample.aux` as well as some other files including `basic-sample.glo` and `basic-sample.ist`. Now load the auxiliary file (`basic-sample.aux`) into MakeGlossariesGUI. (Since the main `.tex` file has the same basename as the `.aux` file, you can also use that, as mentioned above.)

Once MakeGlossariesGUI has successfully loaded the `.aux` file, it will run `makeindex` on the associated `.glo` file (with the `.ist` file as the style). In this case, there are no problems with the document and the `.tex` file is now ready for another L^AT_EX run. The general information panel ([Figure 2.1](#)) shows a summary of the document glossaries. In this case, there's only one glossary (the `main` one). If you edit the document source code

(.tex file) you can use the File→Reload menu item to reload the updated .aux file in MakeGlossariesGUI.

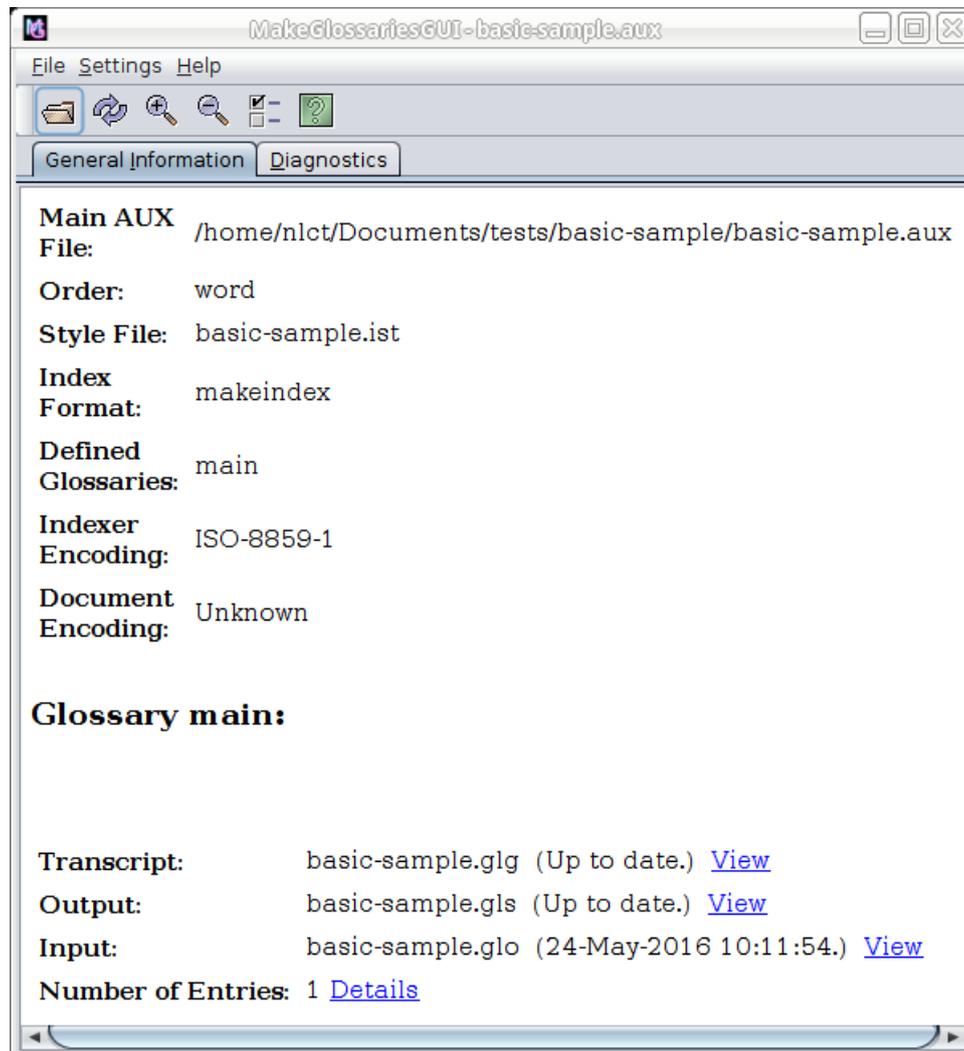


Figure 2.1: General Information Panel (Basic Sample)

In this example, only one entry has been indexed in the `main` glossary. You can find out more information by clicking on the “Details” link, which will open the window shown in [Figure 2.2](#).

Since only one entry has been used, there’s only one row. The first column lists the entry’s label, the second column lists the entry’s sort field and the third column shows the number of times that entry was indexed in the document. If you have a long list of entries, you can use the search box to find an entry according to its *label*. (The sort column isn’t searched.) Regular expressions are permitted.

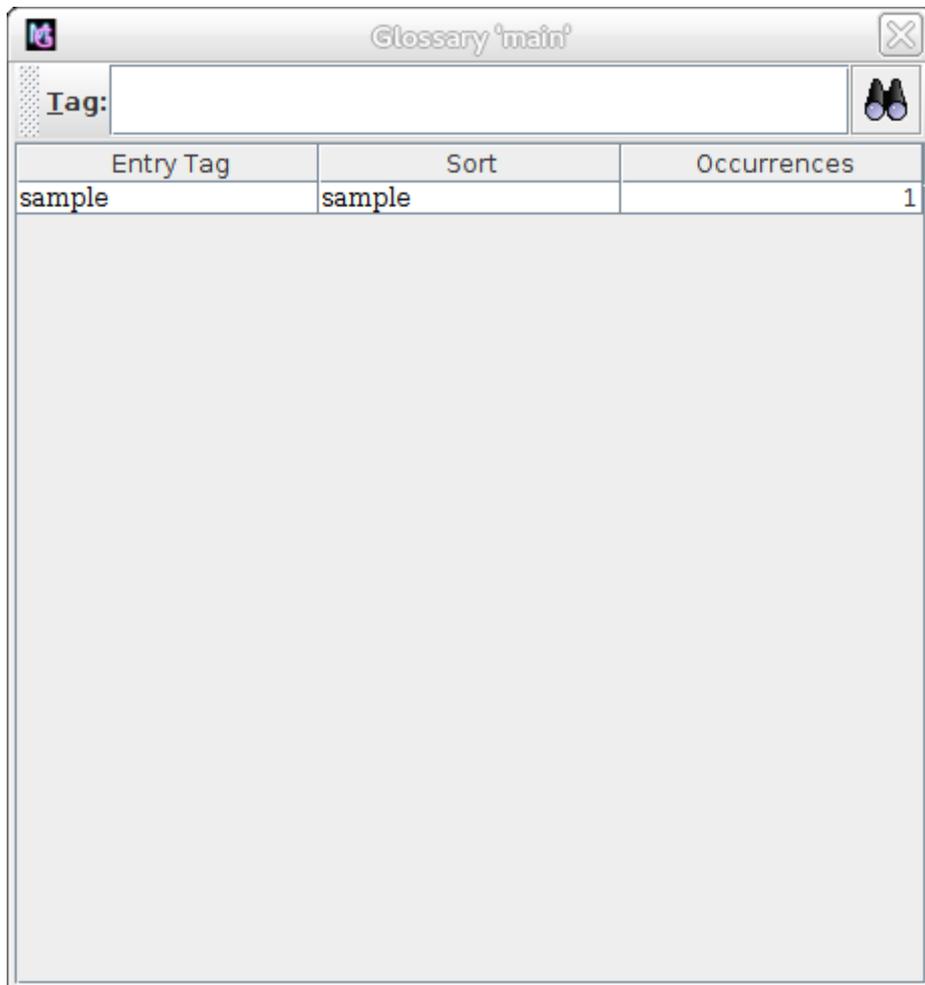


Figure 2.2: Entry Details (Basic Sample)

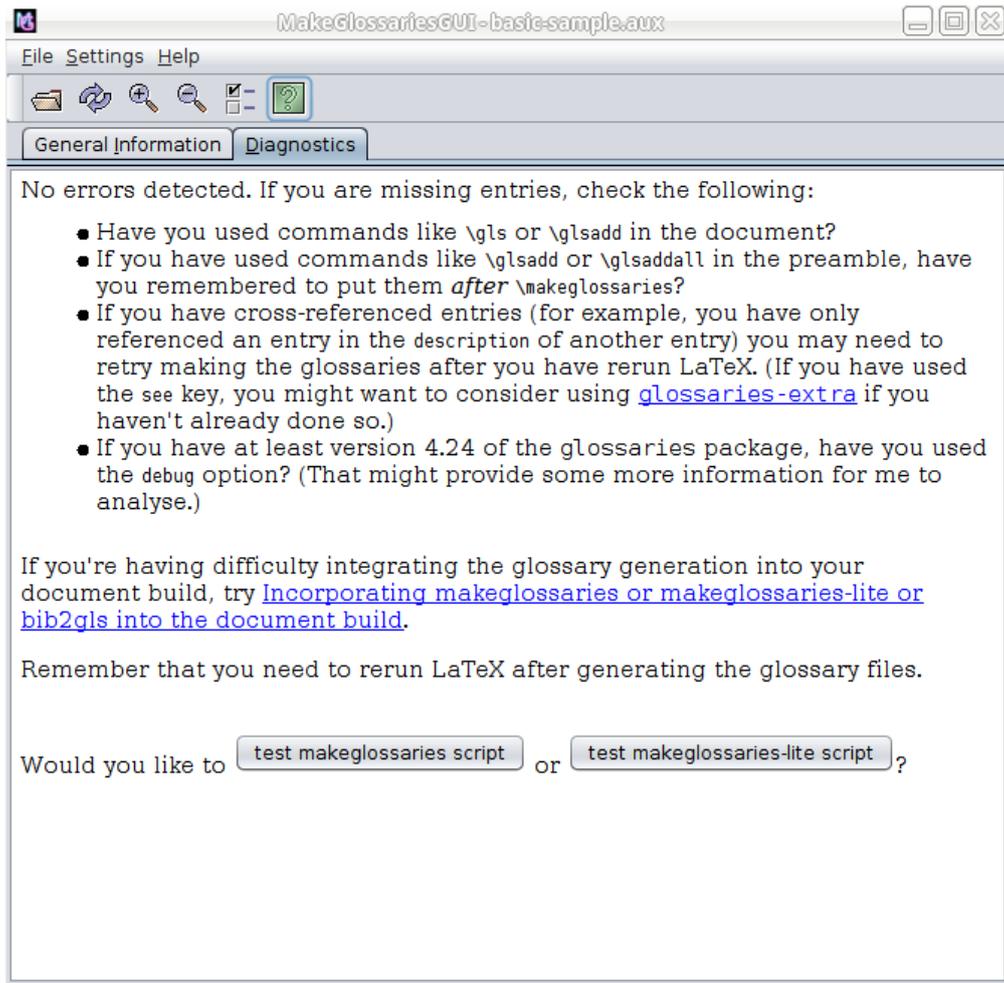


Figure 2.3: Diagnostics Tab (Basic Sample)

The **Diagnostics** tab (Figure 2.3) provides information, warnings and suggestions. In this example, there are no errors detected, so it just provides suggestions and some links on how to incorporate `makeglossaries` into your document build process. There are also two buttons provided to test the `makeglossaries` and `makeglossaries-lite.lua` scripts. In the first case, the action will also test if Perl is installed.

If you have defined an entry in your document, but it's not listed in the details window for the relevant glossary, then it hasn't been indexed in your document. Remember that the commands described in section 9 ("Using Glossary Terms Without Links") of the glossaries manual don't index the terms. These essentially are all the commands in the form `\glsentry<field>` or `\glossentryname<field>`, such as `\glsentrytext`, `\glsentryshort`, `\glsentrylong` or `\glossentryname`, and their case-changing vari-

ants. Also `\glsentrytitlecase` and `\gls hyperlink`.

If you're using the `glossaries-extra` package, remember that the `noindex` option will suppress indexing.

Example 2.

Now let's consider the following document (called, say, `missing-sort.tex`):

```
\documentclass{article}

\usepackage[utf8]{inputenc}
\usepackage[xindy]{glossaries}

\makeglossaries

\newglossaryentry{S}{name={\S},
  description={section symbol}}

\newglossaryentry{alpha}{name={\ensuremath{\alpha}},
  description={alpha}}

\newglossaryentry{beta}{name={\beta},text={\beta},
  description={beta}}

\begin{document}
Test: \gls{S}, $\gls{alpha}$, $\gls{beta}$.

\printglossaries

\end{document}
```

As before, run \LaTeX as usual on this document. Since the `xindy` package option has been used, this will create a `.xdy` file instead of a `.ist` file and the `.glo` file is now in `xindy`'s format. There are, however, problems with this document. The `glossaries` manual advises using the `sort` key for entries that contain special characters or commands in the entry's name. This document hasn't followed that advice, and `xindy` will complain. The `S` entry just causes a warning:

Would replace complete index key with empty string, ignoring
and the `S` entry is ignored. The `alpha` and `beta` entries cause an error:

index 0 should be less than the length of the string

Again the entries are ignored, but the message is fairly cryptic. If we load the auxiliary file (`missing-sort.aux`) into `MakeGlossariesGUI`, these problems are detected, and the following error message is displayed:

Xindy has ignored one or more entries with empty sort strings. Xindy failed with exit code 1.

Once this error message has been dismissed, the **Diagnostics** tab should automatically be selected (see [Figure 2.4](#)). This identifies the problem entries and recommends a solution, in this case, add the `sort` key to the entry definition. The actual warning and error message reported by `xindy` are shown at the end. (You can adjust the font used by these messages if you like, see [chapter 3](#).)

In the **General Information** panel, the “Details” link can again be used to view the list of indexed entries, but now the problematic entries are shown in red (see [Figure 2.5](#)).

MakeGlossariesGUI actually performs more problem-checking than `makeglossaries` as it also tries to parse the log file for certain messages (but only in **GUI** mode). This is illustrated in the next example, which won’t generate any error messages from `makeglossaries`.

Example 3.

Spot what’s wrong with the following document:

```
\documentclass{article}

\usepackage{glossaries}

\makeglossaries

\newglossaryentry{sample}{name={sample},
description={an example}}

\begin{document}
A \gls{sample} document.

\printglossary[type=acronym]

\end{document}
```

If you use the normal method of \LaTeX , `makeglossaries`, \LaTeX , you won’t get any error messages, but the glossary won’t be displayed. Why not? If we switch from

```
\usepackage{glossaries}
```

to

```
\usepackage{glossaries-extra}
```

then we do finally get an error:

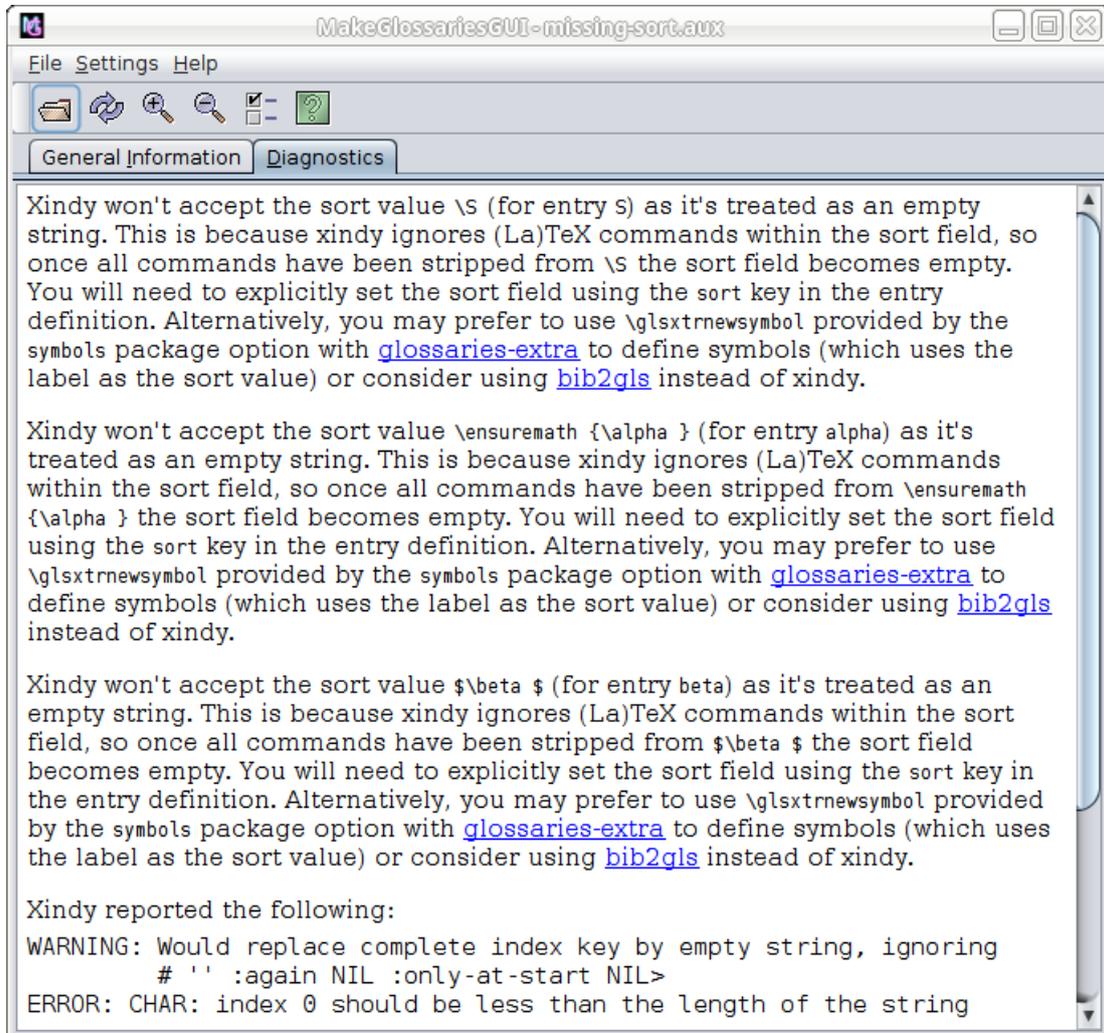


Figure 2.4: Diagnostics Panel

The screenshot shows a window titled "Glossary 'main'" with a search bar labeled "Tag:" and a magnifying glass icon. Below the search bar is a table with the following data:

Entry Tag	Sort	Occurrences
S	\S	1
beta	β	1
alpha	α	1

Figure 2.5: Entry Details (Problematic Entries)

! Package glossaries-extra Error: Glossary type 'acronym' doesn't exist.

The `glossaries-extra` package is stricter than the base `glossaries` package. The problem here is that I've used `type=acronym`, but there's no glossary with that label. (I haven't used the `acronym` option.) If you're not using the extension package, this is harder to pick up, but `MakeGlossariesGUI` will notify you of the problem. This example document will trigger the error

No glossary 'acronym'.

and the diagnostics panel will show the message:

It looks as though you might have done something like
`\printglossary[type={acronym}]`, but there's no acronym glossary.

If you switch to the **General Information** panel, the labels for the glossaries defined in the document are listed next to "Defined Glossaries:" so you can check the indicated type against it. In this example, the list has only the single label "main".

Remember that you not only have to define your entries, but you also have to index them if you want them to appear in the glossary. The `glossaries` package provides many commands that index entries, the most commonly used one being `\gls`, which displays the text associated with the entry, indexes the entry, marks it as having been used and (if the `hyperref` package has been loaded) also creates a link to the definition in the glossary. Other commands provide variations, such as displaying different text or not changing the "first use flag". In particular, the `\glsadd` command only indexes the entry without displaying any text.

Example 4.

In the sample document below, I've defined an entry but it hasn't been indexed anywhere in the document.

```
\documentclass{article}

\usepackage{glossaries}

\makeglossaries

\newglossaryentry{sample}{name={sample},
  description={an example}}

\begin{document}
A sample document.

\printglossaries

\end{document}
```

My first step, as usual, is to run L^AT_EX on this document, which will create the .aux file. Now if I try loading this file into MakeGlossariesGUI I get the error message:

No entries were found for glossary 'main'.

The diagnostics panel shows the following message:

There were no entries listed for the main glossary. Remember that you must index entries for them to appear in the glossary using the commands provided by the glossaries package. Entries that have been defined but not indexed won't be listed. If you don't want to use this glossary, add the `nomain` package option to your document. Check the following:

- Have you used commands like `\gls` or `\glsadd` in the document? (If you haven't, you need to add them.)
- If you have used commands like `\glsadd` or `\glsaddall` in the preamble, have you remembered to put them *after* `\makeglossaries`
- If you have at least version 4.24 of the glossaries package, have you used the debug option? (That might provide some more information for me to analyse.)

(The sentence referencing `nomain` only appears if there are no entries for the main glossary, but not for any other glossaries.)

Remember that if you use `\makenoidxglossaries`, you don't need `makeindex` or `xindy`.

Example 5.

Consider the following document:

```
\documentclass{article}

\usepackage{glossaries}

\makenoidxglossaries

\newglossaryentry{sample}{name={sample},
description={an example}}

\begin{document}
A \gls{sample} document.

\printnoidxglossaries

\end{document}
```

If I load the .aux file for this document into MakeGlossariesGUI, I get the following message in the diagnostics panel:

```
It seems you've used \makenoidxglossaries, which means you don't need
xindy or makeindex, you just need a second LaTeX run to get the glossary up
to date.
```

Note that MakeGlossariesGUI can still provide some limited diagnostics even when `\makenoidxglossaries` has been used. To illustrate this, if we modify the above sample document slightly, introducing an error:

```
\documentclass{article}

\usepackage{glossaries}

\makenoidxglossaries

\newglossaryentry{sample}{name={sample},
description={an example}}

\begin{document}
A \gls{sample} document.

\printnoidxglossary[type=acronym]

\end{document}
```

This provides some additional information in the diagnostics panel:

```
Package glossaries Warning: Empty glossary for
\printnoidxglossary[type={acronym}] Rerun may be required (or you
may have forgotten to use commands like \gls) on input line 13.

It looks as though you might have done something like
\printnoidxglossary[type={acronym}], but there's no acronym glossary.
```

So MakeGlossariesGUI picks up the error.

Note that MakeGlossariesGUI also looks for warnings from the `glossaries` package, so if you are encountering any problems, make sure you haven't suppressed the warnings with the `nowarn` package option.

Example 6.

In this example I've omitted `\printglossary` from the document:

```

\documentclass{article}

\usepackage{glossaries}

\makeglossaries

\newglossaryentry{sample}{name={sample},
description={an example}}

\begin{document}
A \gls{sample} document.

\end{document}

```

This doesn't cause any problems for `makeindex` as all the associated files have been created correctly. The document simply doesn't load the file generated by `makeindex` as there's no `\printglossary` (or `\printglossaries`). However the `glossaries` package does generate a warning, and this warning is picked up by `MakeGlossariesGUI` and displayed in the diagnostics panel:

```

Package glossaries Warning: No \printglossary or \printglossaries
found. (Remove \makeglossaries if you don't want any glossaries.) This
document will not have a glossary.

```

If you suppress these warnings then `MakeGlossariesGUI` can't help.

Sometimes things can go so badly wrong that `LATEX` doesn't even generate an auxiliary file. In this case you can load the `.log` file instead. (You'll need to change the file selector filter to show all files.) `MakeGlossariesGUI` will parse the log file to see if it recognises any of the error messages. Some `LATEX` error messages can be quite cryptic so there's no guarantee that `MakeGlossariesGUI` will be able to help, but it might detect something useful. Note that this option is only available in `GUI` mode.

Example 7.

Consider the following document:

```

\batchmode
\documentclass{beamer}

\usepackage{glossaries}

\makeglossaries

\newglossaryentry{sample}{name=sample,first={\textit{sample}},
description={an example}}

```

```

\begin{document}

\begin{frame}
\gls{sample}
\end{frame}

\begin{frame}
\printglossary
\end{frame}

\end{document}

```

This document goes badly wrong. The first error message is:

```

! Undefined control sequence.
\in@ #1#2->\begingroup \def \in@@

```

If I load the log file into MakeGlossariesGUI, the diagnostic panel displays the following:

Since the aux file doesn't exist, there's not much I can do to help, but I'll parse the log file in case there are any clues there.

It's possible that there's an expansion issue involving a fragile command. Things to check for:

- Have you used a class like `beamer` that doesn't make common formatting commands like `\textit` robust?
- Have you tried using `\protect` in front of fragile commands contained within your entry definitions?
- Have you tried switching off the expansion using commands like `\glsnoexpandfields`? (See section 4.6 Expansion in the `glossaries` user manual.)

The problem here is that a fragile command has been used in the entry definition. The problematic command in this example is `\textit`, which is normally robust, but it happens to be fragile with the `beamer` class. The solution is to either protect the problematic command with `\protect` or use `\glsnoexpandfields` before you define the entries. For example:

```

\documentclass{beamer}

\usepackage{glossaries}

\makeglossaries

```

```

\glsnoexpandfields

\newglossaryentry{sample}{name=sample,first={\textit{sample}},
  description={an example}}

\begin{document}

\begin{frame}
\gls{sample}
\end{frame}

\begin{frame}
\printglossary
\end{frame}

\end{document}

```

The main panel shows the character encoding that MakeGlossariesGUI believes is being used by the indexing application and, if detected, the document input encoding. Since `makeindex` only supports characters in the range 1 to 255, MakeGlossariesGUI assumes ISO-8859-1 (Latin-1) for `makeindex` and will add an advisory note if the document class uses a different encoding.

Example 8.

Consider the following example:

```

\documentclass{article}

\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}

\usepackage[style=indexgroup]{glossaries}

\makeglossaries

\newglossaryentry{elite}{name={\acute{e}lite},description={select group}}

\begin{document}
\gls{elite}

\printglossary
\end{document}

```

This document is saved as UTF-8 and has a term where the sort value starts with

an extended character. This doesn't work as `makeindex` treats the UTF-8 letter as two separate characters from the two octets. This not only affects the sorting but also causes a problem for the `indexgroup` style.

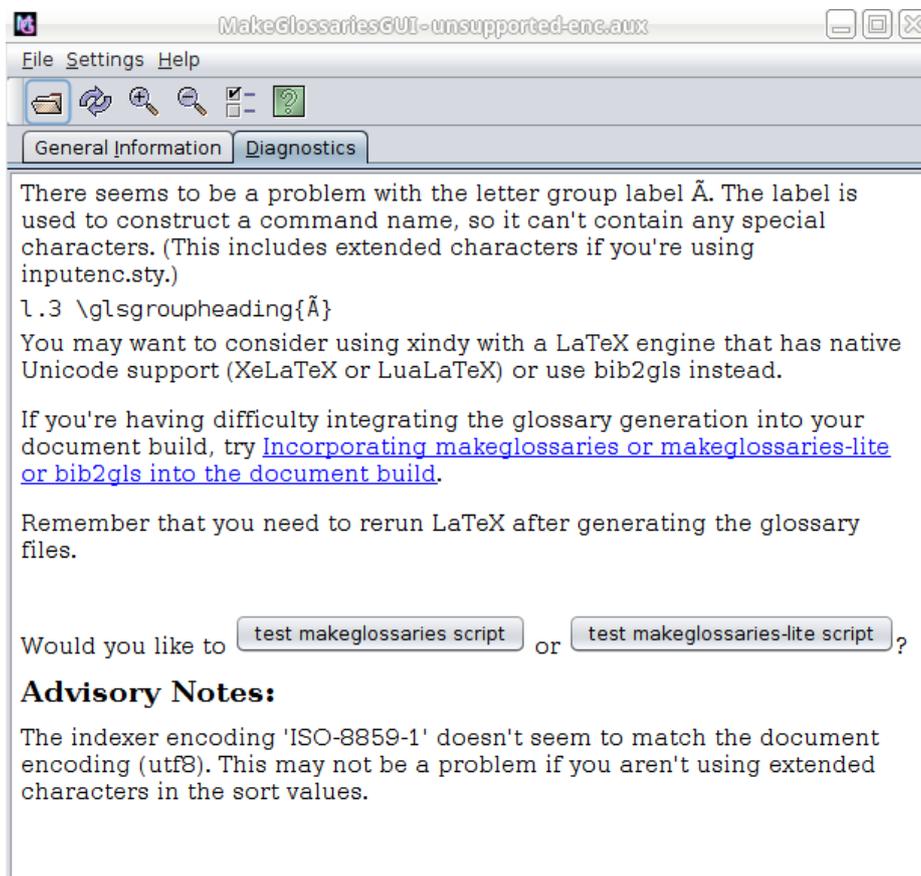


Figure 2.6: Encoding Problems (Diagnostics Panel)

The first message in the diagnostics panel (see [Figure 2.6](#)) is only picked up after you rerun \LaTeX and reload the file in `MakeGlossariesGUI`, as it's only after the glossary file has been created by `makeindex` that the \LaTeX call fails. The failure is caused by the first octet appearing in the argument of `\glsgroupheading`. This causes two problems: the argument of this command is a label so special or active characters will break it, and the `inputenc` package makes the first octet active, requiring the second octet as the argument. The message reads:

There seems to be a problem with the letter group label \tilde{A} . The label is used to construct a command name, so it can't contain any special characters. (This includes extended characters if you're using `inputenc.sty`.)

```
1.3 \glsgroupheading{\tilde{A}}
```

You may want to consider using xindy with a LaTeX engine that has native Unicode support (XeLaTeX or LuaLaTeX) or use bib2gls instead.

This message doesn't show up when you first attempt to create the glossary files with MakeGlossariesGUI. However, there's advisory message than points to a problem:

The indexer encoding 'ISO-8859-1' doesn't seem to match the document encoding (utf8). This may not be a problem if you aren't using extended characters in the sort values.

In this case it is a problem. The two different encodings are also shown in [Figure 2.6](#). The indexer encoding is listed as ISO-8859-1, and the document encoding is listed as utf8.

The problem shows up more clearly in the "Details" window (see [Figure 2.8](#)), which uses the indexer's encoding and so displays the sort value as Ã©lite instead of élite.

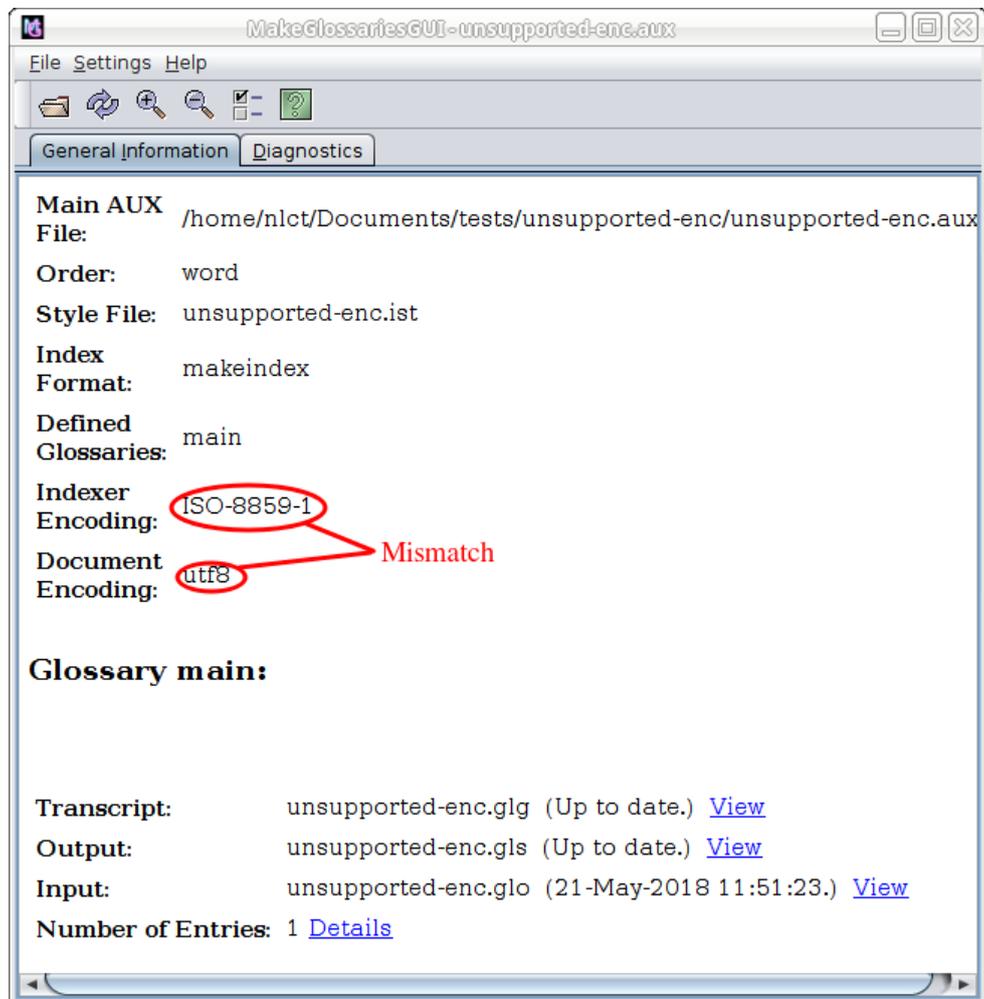


Figure 2.7: Encoding Problems (General Information Panel)

The image shows a window titled "Glossary 'main'". At the top, there is a search bar labeled "Tag:" with a magnifying glass icon to its right. Below the search bar is a table with three columns: "Entry Tag", "Sort", and "Occurrences". The table contains one row with the following data:

Entry Tag	Sort	Occurrences
elite	A@lite	1

The rest of the window area is a large, empty grey rectangle.

Figure 2.8: Encoding Problems (Details Window)

3 Settings

The application settings can be adjusted through the Settings menu. This has menu items for increasing or decreasing the font size (Settings→Increase Font Size or Settings→Decrease Font Size), setting the dry run mode (Settings→Dry Run Mode) or open the **Preferences** dialog window (Settings→Edit Properties). Note that the dry run mode is the only setting that isn't remembered the next time you run MakeGlossariesGUI.

The **Preferences** dialog has four tabs: **Start Up Directory**, **Diagnostics**, **Applications** and **GUI**.

3.1 Start Up Directory

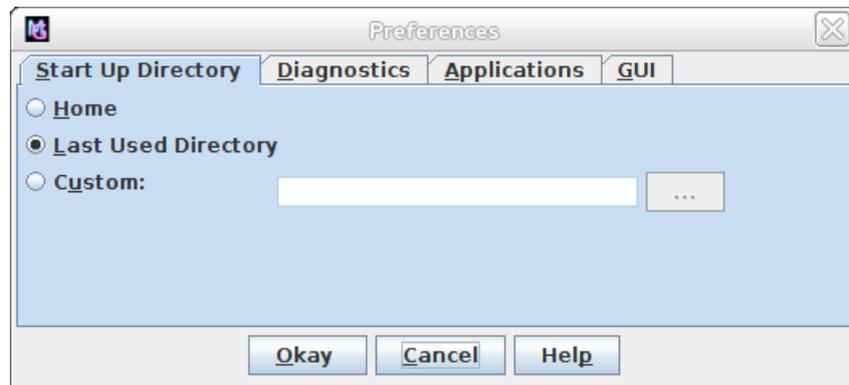


Figure 3.1: Properties Dialog (Start Up Directory)

The **Start Up Directory** (Figure 3.1) tab allows you to select the directory to use on start up. This is the directory the file chooser will be set to initially.

3.2 Diagnostic Settings

MakeGlossariesGUI will try to determine if you have defined any entries within the document environment. Although the glossaries package allows document definitions, the manual encourages defining entries in the preamble, see section 4.10 of the glossaries user manual (“Drawbacks With Defining Entries in the Document Environment”). If you want to skip this check, deselect the **Check for document definitions** check box.

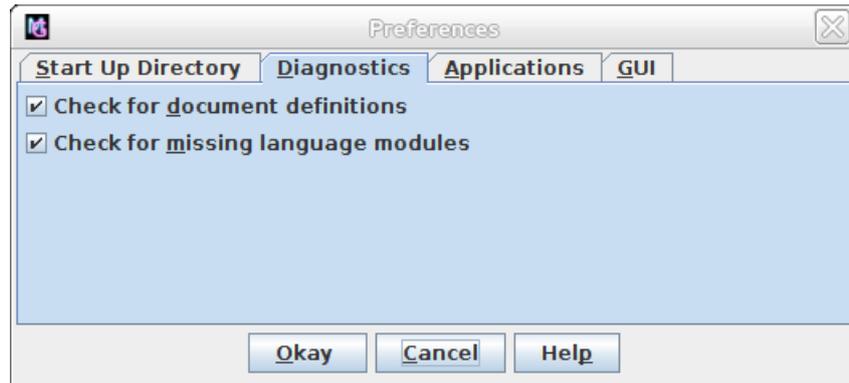


Figure 3.2: Properties Dialog (Diagnostics)

There’s also a check to see if the `glossaries` package has complained about missing language modules. Not all languages are supported and, for those languages that are supported, the appropriate module must be installed in addition to installing the `glossaries` package. If the required language support is missing, the glossary files can still be built, you’ll just have to manually change the fixed text for the title following the instructions in section 1.3 (“Multi-Lingual Support”) of the `glossaries` user manual. If you want to skip this check, deselect the **Check for missing language modules** check box.

3.3 Indexing Applications

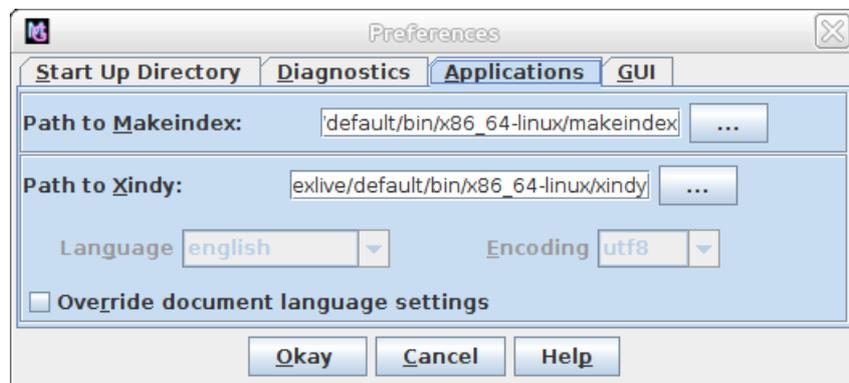


Figure 3.3: Properties Dialog (Applications)

The **Applications** tab lists the paths to `makeindex` and `xindy`. `MakeGlossariesGUI` will attempt to locate them on your system’s path, but if they can’t be detected, you’ll need

to specify the correct location. You can omit the location for an unrequired application.

MakeGlossariesGUI will try to determine the language and input encoding from the `.aux` file to pass to `xindy`, but you can override this if you want to. Make sure that the **Override document language settings** check box is selected, and change the language and encoding as appropriate. Note that the batch mode will also use these settings, although they can only be adjusted in the **GUI** mode.

3.4 GUI Preferences

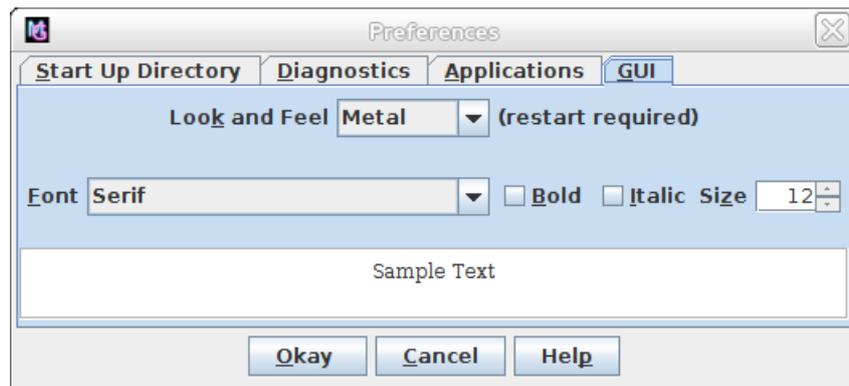


Figure 3.4: GUI Preferences (Metal Look and Feel)

The font used in the **General Information** and **Diagnostics** panels can be set in the **GUI** tab. In addition to adjusting the font size through the **Settings**→**Increase Font Size** or **Settings**→**Decrease Font Size** menu items, you can also set the required font size in this tab.

The **Look and Feel** can be set by selecting the required option in the **Look and Feel** drop-down menu. Note that a restart is required as the Look and Feel is set on start up. The title bar appearance is governed by your usual operating system preference. The Look and Feel changes the way the window elements are displayed. **Figure 3.4** shows the “Metal” Look and Feel. **Figure 3.5** shows the “Nimbus” Look and Feel. **Figure 3.6** shows the “CDE/Motif” Look and Feel. **Figure 3.7** shows the “GTK+” Look and Feel. You may not have all these options on your system or you may have additional options, depending on your Java installation.

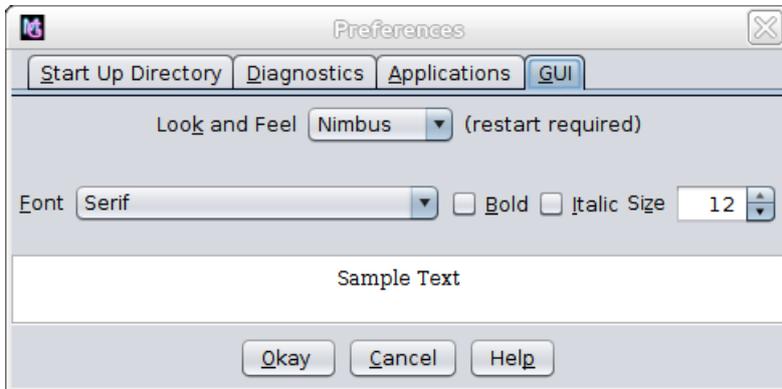


Figure 3.5: GUI Preferences (Nimbus Look and Feel)

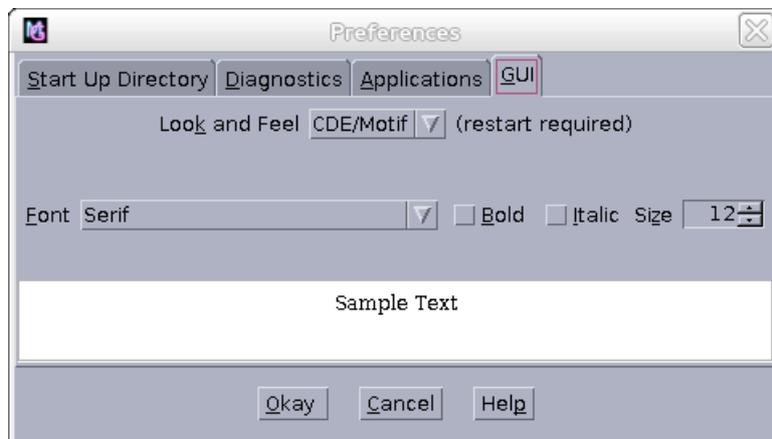


Figure 3.6: GUI Preferences (CDE/Motif Look and Feel)

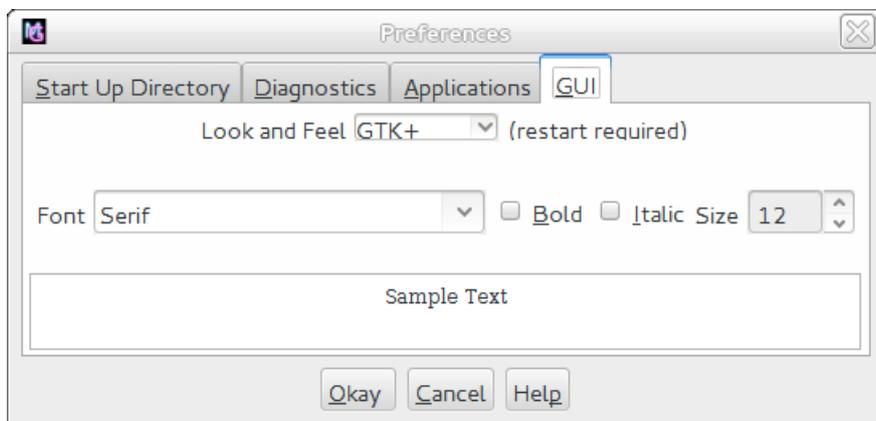


Figure 3.7: GUI Preferences (GTK+ Look and Feel)

4 Licence

MakeGlossariesGUI is licensed under the terms of the GNU General Public License. MakeGlossariesGUI depends on the following third party libraries whose jar files are in the lib directory: Java Help (<https://javahelp.java.net/>) and the Java Look and Feel Graphics Repository (<http://www.oracle.com/technetwork/java/index-138612.html>).

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 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) <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>.

Glossary

bib2gls an indexing application designed specifically for use with glossaries-extra.sty. [3](#)

GUI graphical user interface. [3–5](#), [10](#), [16](#), [25](#)

Look and Feel the appearance (look) of Java applications and how the widgets behave (feel). [25](#)

makeglossaries a Perl script provided with the glossaries package which runs either makeindex or xindy, depending on the settings in the .aux file. [3](#), [4](#), [8](#), [10](#)

makeglossaries-lite.lua a Lua script provided with the glossaries package as a light-weight alternative to makeglossaries (the .lua extension may be omitted or replaced with .exe, according to the TeX distribution). [3](#), [8](#)

makeindex an indexing application. [3–5](#), [14](#), [16](#), [18](#), [19](#), [24](#)

xindy an indexing application written in Perl. [3](#), [4](#), [9](#), [10](#), [14](#), [24](#), [25](#)

Index

- `--batch`, 3, 4
- `--debug`, 4
- `--dry-run`, 4
- `--gui`, 4
- `--help`, 4
- `--nodry-run`, 4
- `--quiet`, 4
- `--version`, 4
- `bib2gls`, 44
- document environment, 23
- File
 - Open, 5
 - Reload, 6
- glossaries package, 3, 8, 9, 13, 15, 16, 23, 24
- glossaries-extra package, 3, 9, 13
- `\glossentryname`, 8
- `\gls`, 13–15
- `\glsadd`, 13, 14
- `\glsaddall`, 14
- `\glsentrylong`, 8
- `\glsentryshort`, 8
- `\glsentrytext`, 8
- `\glsentrytitlecase`, 9
- `\glsgroupheading`, 19
- `\glshyperlink`, 9
- `\glsnoexpandfields`, 17
- hyperref package, 13
- inputenc package, 19
- Look and Feel, 44
- `\makeglossaries`, 14, 16
- `makeglossaries`, 44
- `makeglossaries-lite.lua`, 44
- `makeindex`, 44
- `\makenoidxglossaries`, 14, 15
- `\printglossaries`, 16
- `\printglossary`, 13, 15, 16
- `\printnoidxglossary`, 15
- `\printunsrtglossary`, 3
- `\protect`, 17
- Settings, 23
 - Decrease Font Size, 23, 25
 - Dry Run Mode, 23
 - Edit Properties, 23
 - Increase Font Size, 23, 25
- `\textit`, 17
- `xindy`, 44