

The glossaries package: a guide for beginners

Nicola L.C. Talbot

2014-01-20

Abstract

This document is a brief guide to the glossaries package for beginners who find the size of the main user manual daunting and, as such, it only covers the basics. For more advanced topics, see the main user manual ([glossaries-user.pdf](#)).

Contents

1 What Do I Need?	1
2 Getting Started	2
3 Defining a Term or Symbol	2
4 Using a Term	4
5 Acronyms	5
6 Displaying the Glossary or List of Acronyms	6

1 What Do I Need?

You need to have the glossaries package installed. This and all dependent packages are contained in both the TeX Live and MikTeX distributions. Use your distribution's update manager to ensure you have an up-to-date version. These packages can also be downloaded from CTAN if you don't have them installed. See "Installing things on a (La)TeX System"¹ for further assistance.

¹<http://www.tex.ac.uk/cgi-bin/texfaq2html?label=installthings>

In addition, you need an indexing application. The glossaries package is configured to use either `makeindex` or `xindy`. `Makeindex` should be available on all TeX systems. It's possible you may not have `xindy` installed, but it can be downloaded from the `xindy` home page². MiKTeX users may find the `Xindy for MiKTeX`³ thread on the LaTeX Community Forum helpful.

The glossaries package provides a helper Perl script called `makeglossaries` which will act as an interface to `makeindex` or `xindy`. If you want to use `makeglossaries` or `xindy`, you need to have Perl installed. If you don't have Perl installed, you can still use `glossaries` but you can only use it with `makeindex`.

2 Getting Started

The most obvious things to do is to remember to load the glossaries package:

```
\usepackage{glossaries}
```

Note that the `glossaries` package must be loaded *after* the `hyperref` package (contrary to the general advice that `hyperref` should be loaded last). The `glossaries` package should also be loaded after `html`, `inputenc`, `babel` and `ngerman`. If you are using `babel` I recommend you use the `glossaries` package option `translate=babel`:

```
\usepackage{babel}  
\usepackage[translate=babel]{glossaries}
```

If you want to use `xindy` rather than `makeindex` to sort the glossaries, you need to specify this in the package options:

```
\usepackage[xindy]{glossaries}
```

Other package options are described in Section 2 in the main `glossaries` user manual.

Next, you need to enable the creation of the various external files that are required by the indexing application. This is done using:

```
\makeglossaries
```

Don't attempt to input any of these external files in your document.

²<http://www.xindy.org/>

³<http://www.latex-community.org/forum/viewtopic.php?f=51&t=5383>

3 Defining a Term or Symbol

You must define terms or symbols before you can use them in the document. This is best done in the preamble. Each term is defined using:

```
\newglossaryentry{<label>}{<settings>}
```

where *<label>* is a unique label used to identify the term. The second argument, *<settings>*, is a key=value comma separated list that is used to set the required information for the term. A full list of available keys can be found in Section 4 in the main glossaries user manual. The principle keys are name and description.

For example, to define the term “electrolyte”:

```
\newglossaryentry{electrolyte}{name=electrolyte,  
description={solution able to conduct electric current}}
```

In the above example, the label and the name happen to be the same. In the next example, the name contains a ligature but the label doesn't:

```
\newglossaryentry{oesophagus}{name=\oe sophagus,  
description={canal from mouth to stomach},  
plural=\oe sophagi}
```

In this case the label is `oesophagus`. Note that in this case, the plural has to be specified using the plural key since it isn't formed by simply appending the letter “s”.

If you want to define all your terms in a separate file *don't* use `\include` to input the file. You can use `\input` but it's better to use `\loadglsentries`. (See Section 4.6.)

When you define terms, it is important to remember that they will be sorted by an external application that doesn't interpret \LaTeX commands. `xindy` is a little more \TeX -aware than `makeindex`, but for this example they will both mis-sort `\oe sophagus`. `Xindy` will ignore the `\oe_` part and will sort it according to `sophagus`, so it will be placed with the terms starting with an “s”. `makeindex`, on the other hand, will sort it according to `\oe_sophagus` which starts with a symbol (the backslash), so it will end up in the symbol group. You therefore need to specify how to sort the term:

```
\newglossaryentry{oesophagus}{name=\oe sophagus,  
description={canal from mouth to stomach},  
plural=\oe sophagi,  
sort=oesophagus}
```

Note, however, that if you use the `inputenc` package and `xindy`, you can use the encoded character `œ` rather than `\oe` and `xindy` will sort it correctly:

```
\newglossaryentry{oesophagus}{name=œsophagus,
description={canal from mouth to stomach},
plural=œsophagi}
```

If the name starts with an accented character, the first character must be grouped:

```
\newglossaryentry{elite}{name={\'}elite,
description={select group of people},
sort=elite}
```

or

```
\newglossaryentry{elite}{name={é}lite,
description={select group of people}}
```

The name can also be a symbol, but you will also need to specify the sort key. For example:

```
\newglossaryentry{pi}{name={\ensuremath{\pi}},
description={ratio of circumference of circle to its
diameter}, sort=pi}
```

It's also possible to have a name and a symbol. For example:

```
\newglossaryentry{set}{name=set,
description={a collection of objects},
symbol={\ensuremath{\mathcal{S}}}}
```

Note, however, that not all glossary styles display the symbol in the glossary.

4 Using a Term

Once you have defined a term, you can use it in the document. The glossaries package provides a number of commands that are described in Section 6 in the main glossaries user manual. Here, I shall just cover the main commands:

```
\gls{label}
```

This prints the term associated with *label*. Using the example in the previous section, `\gls{oesophagus}` will display œsophagus. If the `hyperref` package has also been loaded, the term will also be hyperlinked to the relevant entry in the glossary.

```
\glspl{<label>}
```

This prints the plural of the term associated with $\langle label \rangle$, so `\glspl{oesophagus}` will display *oesophagi*. Again, the text will be a hyperlink to the relevant entry in the glossary, if hyperlinks are defined.

There are also versions that convert the first character to upper case if you need to start a sentence with a term:

```
\Gls{<label>}
```

and

```
\Glspl{<label>}
```

For example:

```
\Glspl{electrolyte} usually exist as solutions of salts,  
bases or acids.
```

Note that these commands all have two optional arguments that aren't discussed here, but they are described in Section 6 in the main glossaries user manual.

Don't use these commands in moving arguments, such as those used by `\chapter`, `\section` and `\caption`.

You can, instead, use commands such as

```
\glsentrytext{<label>}
```

See Section 9 in the main glossaries user manual.

5 Acronyms

The glossaries package also allows you to define acronyms. By default, acronyms will be added to the main glossary, but if you use the acronym package option, acronyms will be placed in a separate list of acronyms:

```
\usepackage[acronym]{glossaries}
```

Acronyms must be defined before use and are defined using:

```
\newacronym{<label>}{<abbrv>}{<full>}
```

where $\langle label \rangle$ is the label identifying the acronym, $\langle abbrev \rangle$ is the abbreviated form and $\langle full \rangle$ is the fully expanded text. For example:

```
\newacronym{label}{svm}{support vector machine}
```

Once the acronym is defined, it can be used in the same way as any other glossary term with commands such as `\gls` and `\glspl`. On first use, the default will display $\langle full \rangle$ ($\langle abbrev \rangle$). On subsequent use, only the abbreviation will appear. For example:

```
A \gls{svm} was implemented.
```

If this is the first time this term has been used, the above would produce:

A support vector machine (svm) was implemented.

otherwise it would produce:

A svm was implemented.

If you prefer a different way of displaying the acronyms (for example, putting the long form in a footnote on first use) it's possible to change the acronym style. For further details, see Section 13 in the main glossaries user manual.

You can reset an acronym so that the next use fully expands it again using:

```
\glsreset{<label>}
```

or you can reset all acronyms using

```
\glsresetall
```

For further details on resetting and unsetting acronyms, see Section 14 in the main glossaries user manual.

6 Displaying the Glossary or List of Acronyms

In order to display the sorted list of terms, you first need to use

```
\printglossaries
```

at the place where you want the glossary and list of acronyms to appear. Note that `\printglossaries` won't produce any text the first time you create your document, as an external indexing application needs to be

used to sort and collate the entries. Creating the completed document is therefore *at least* a three step process.

Suppose your document is called `myDoc.tex`, then you need to:

1. Build/L^AT_EX your document, either by clicking on an appropriate button in your front-end/text editor or by typing the following at a terminal/command prompt:

```
latex myDoc
```

2. Run the indexing application on the external file(s) created by the glossaries package. If you have used the `xindy` package option, you must use `xindy`, otherwise you must use `makeindex`. The glossaries package comes with a convenient Perl wrapper that calls the appropriate indexing application the required number of times with the required options. If you have Perl installed, all you need to do is type

```
makeglossaries myDoc
```

in a terminal or command prompt. If you are using a front-end (such as WinEdt or TeXnicCenter) and you want it to automatically run `makeglossaries` when you build your document, you will need to consult the front-end's documentation.

If you don't have Perl installed, you will need to call `makeindex` or `xindy` explicitly. This is more complicated and is described in Section 1.3 in the main glossaries user manual.

3. Once you have successfully run the indexing application, you need to build/L^AT_EX your document again (see step 1).

Note that sometimes you may have to repeat steps 2 and 3.

The appearance of the glossary depends on the glossary style in use. The default style uses the `description` environment, but there are many other styles available. These are described in Section 15 in the main glossaries user manual. By default, a number or list of numbers will appear after each entry in the glossary. This indicates the page (or pages) on which the term was used. This list can be suppressed using the `nonumberlist` package option:

```
\usepackage[nonumberlist]{glossaries}
```

Further information can be found in the main glossaries user manual ([glossaries-user.pdf](#)) and there is also an [article on the glossaries package](#) on the L^AT_EX Community's⁴ Know How section.

⁴<http://www.latex-community.org/>