Upgrading from the glossary package to the glossaries package

Nicola L.C. Talbot

March 8, 2008

Contents

Abstract

The purpose of this document is to provide advice if you want to convert a LaTeX document from using the obsolete glossary package to the replacement glossaries package.

1 Package Options

When converting a document that currently uses the obsolete glossary package to the replacement glossaries package, it should be fairly obvious that the first thing you need to do is replace `\usepackage{glossary}` with `\usepackage{glossaries}`, however some of the package options are different, so you may need to change those as well. Table 1 shows the mappings from the glossary to the glossaries package options.

Table 1: Mappings from glossary to glossaries package options

glossary option	**glossaries option**
style=list	style=list
style=altlist	style=altlist
style=long,header=none,border=none,cols=2	style=long
style=long,header=plain,border=none,cols=2	style=longheader
style=long,header=none,border=plain,cols=2	style=longborder
style=long,header=plain,border=plain,cols=2	style=longheaderborder
style=long,header=none,border=none,cols=3	style=long3col
style=long,header=plain,border=none,cols=3	style=long3colheader
style=long,header=none,border=plain,cols=3	style=long3colborder
style=long,header=plain,border=plain,cols=3	style=long3colheaderborder
style=super,header=none,border=none,cols=2	style=super
style=super,header=plain,border=none,cols=2	style=superheader
style=super,header=none,border=plain,cols=2	style=superborder
style=super,header=plain,border=plain,cols=2	style=superheaderborder
style=super,header=none,border=none,cols=3	style=super3col
style=super,header=plain,border=none,cols=3	style=super3colheader
style=super,header=none,border=plain,cols=3	style=super3colborder
style=super,header=plain,border=plain,cols=3	style=super3colheaderborder
number=none	nonumberlist
number=<*counter name*>	counter=<*counter name*>
toc	toc
hypertoc	toc
hyper	*no corresponding option*
section=true	section
section=false	*no corresponding option*
acronym	acronym
global	*no corresponding option*

2 Defining new glossary types

If you have created new glossary types, you will need to replace all instances of

```
\newglossarytype[<log-ext>]{<type>}{<out-ext>}{<in-ext>}[<style list>]
\newcommand{\<type>name}{<title>}
```

glossary

with

```
\newglossary[<log-ext>]{<type>}{<out-ext>}{<in-ext>}{<title>}
```

glossaries

in the preamble, and

```
\glossarystyle{<new style list>}
```

glossaries

immediately before `\printglossary[type=<type>]`, if the new glossary requires a different style to the main (default) glossary.

The *<style list>* optional argument can be converted to *<new style list>* using the same mapping given in Table 1.

For example, if your document contains the following:

```
\newglossarytype[nlg]{notation}{not}{ntn}[style=long,header]
\newcommand{\notationname}{Index of Notation}
```

You will need to replace the above two lines with:

```
\newglossary[nlg]{notation}{not}{ntn}{Index of Notation}
```

in the preamble, and

```
\glossarystyle{style=longheader}
```

immediately prior to displaying this glossary new type.

Note that the glossary title is no longer specified using `\<glossary-type>name` (except for `\glossaryname` and `\acronymname`) but is instead specified in the *<title>* argument of `\newglossary`. The short title which is specified in the glossary package by the command `\short<glossary-name>name` is now specified using the `toctitle` key in the optional argument to `\printglossary`.

3 `\make`*<glossary name>*

All instances of `\make`*<glossary name>* (e.g. `\makeglossary` and `\makeacronym`) should be replaced by the single command `\makeglossaries`. For example, if your document contained the following:

```
\makeglossary
\makeacronym
```

then you should replace both lines with the single line:

```
\makeglossaries
```

4 Storing glossary information

With the old glossary package you could optionally store glossary information for later use, or you could simply use `\glossary` whenever you wanted to add information to the glossary. With the new glossaries package, the latter option is no longer available[1]. If you have stored all the glossary information using `\storeglosentry`, then you will need to convert these commands into the equivalent `\newglossaryentry`. If you haven't used `\storeglosentry`, then you'll have a bit more work to do!

The former approach requires substituting all instances of

`\storeglosentry{`<*label*>`}{`<*gls-entry*>`}`

glossary

with

`\newglossaryentry{`<*label*>`}{`<*gls-entry*>`}`

glossaries

This should be fairly easy to do using the search and replace facility in your editor (but see notes below).

If you have used the optional argument of `\storeglosentry` (i.e. you have multiple glossaries) then you will need to substitute

`\storeglosentry[`<*gls-type*>`]{`<*label*>`}{`<*gls-entry*>`}`

glossary

with

`\newglossaryentry{`<*label*>`}{`<*gls-entry*>`,type=`<*gls-type*>`}`

glossaries

The glossary entry information <*gls-entry*> may also need changing. If <*gls-entry*> contains any of `makeindex`'s special characters (i.e. `@ ! " ` or `|`) then they should no longer be escaped with `"` since the glossaries package deals with these characters internally. For example, if your document contains the following:

```
\storeglosentry{card}{name={$"|\mathcal{S}"|$},
description={The cardinality of the set $\mathcal{S}$}}
```

then you will need to replace it with:

[1] mainly because having a key value list in `\glossary` caused problems, but it also helps consistency.

```
\newglossaryentry{card}{name={$|\mathcal{S}|$},
description={The cardinality of the set $\mathcal{S}$}}
```

The `format` and `number` keys available in `\storeglosentry` are not available with `\newglossaryentry`. Note also that the glossary package allowed you to use `\storeglosentry` in the document, but `\newglossaryentry` may only be used in the preamble.

5 Adding an entry to the glossary

The glossary package provided two basic means to add information to the glossary: firstly, the term was defined using `\storeglosentry` and the entries for that term were added using `\useglosentry`, `\useGlosentry` and `\gls`. Secondly, the term was added to the glossary using `\glossary`. This second approach is unavailable with the glossaries package.

5.1 `\useglosentry`

The glossary package allows you to add information to the glossary for a predefined term without producing any text in the document using

`\useglosentry[`<*old options*>`]{`<*label*>`}`

glossary

Any occurrences of this command will need to be replaced with

`\glsadd[`<*new options*>`]{`<*label*>`}`

glossaries

The `format` key in <*old options*> remains the same in <*new options*>, the `number=`<*counter name*> key in <*old options*> should be replaced with `counter=`<*counter name*> in <*new options*>.

5.2 `\useGlosentry`

The glossary package allows you to add information to the glossary for a predefined term with the given text using

`\useGlosentry[`<*old options*>`]{`<*label*>`}{`<*text*>`}`

glossary

Any occurrences if this command will need to be replaced with

`\glslink[`<*new options*>`]{`<*label*>`}{`<*text*>`}`

glossaries

The mapping from <*old options*> to <*new options*> is the same as that given in section 5.1 above.

5.3 \gls

Both the glossary and the glossaries packages define the command `\gls`. In this case, the only thing you need to change is the `number` key in the optional argument to `counter`.

5.4 \glossary

When using the glossaries package, you should not use `\glossary` directly.[2] If, with the old package, you have opted to explicitly use `\glossary` instead of storing the glossary information with `\storeglosentry`, then converting from glossary to glossaries will be more time-consuming, although in the end, I hope you will see the benefits![3] If you have used `\glossary` with the old glossary package, you will instead need to define the relevant glossary terms using `\newglossaryentry` and reference the terms using `\glslink`, `\gls` (or `\glspl` etc).

If you don't like the idea of continually scrolling back to the preamble to type all your `\newglossaryentry` commands, you may prefer to create a new file, in which to store all these commands, and then input that file in your document's preamble. Most text editors and front-ends allow you to have multiple files open, and you can tab back and forth between them.

6 Acronyms

In the glossary package, acronyms were treated differently to glossary entries. This resulted in inconsistencies and sprawling unmaintainable code. The new glossaries package treats acronyms in exactly the same way as normal glossary terms. In fact, in the glossaries package:

```
\newacronym[<options>]{<label>}{<abbrv>}{<long>}
```

glossaries

is a shortcut for:

```
\newglossaryentry{<label>}{type=\acronymtype, name={<abbrv>},
description={<long>}, text={<abbrv>}, first={<long>
(<abbrv>)}, plural={<abbrv>s}, firstplural={<long>s
(<abbrv>s)}, <options>}
```

glossaries

[2] This is because `\glossary` requires the argument to be in a specific format and doesn't use the *<key>=<value>* format that the old glossary package used. The new package's internal commands set this format, as well as escaping any of `makeindex`'s special characters, so although it is still possible to use `\glossary` with the new package, it's not recommended. If you persist in using `\glossary` with the new package, don't complain if things go wrong!

[3] From the user's point of view, using `\glossary` throughout the document is time consuming, and if you use it more than once for the same term, there's a chance extra spaces may creep in which will cause `makeindex` to treat the two entries as different terms, even though they look the same in the document.

This is different to the glossary package which set the `name` key to <*long*> (<*abbrv*>) and allowed you to set a description using the `description` key. If you want your document to remain like this, you can redefine `\newacronym` as follows:

```
\renewcommand{\newacronym}[4][]{%
\newglossaryentry{#2}{type=\acronymtype,
name={#4 (#3)},
text={#3},
first={#4 (#3)},
plural={#3s},
firstplural={#4s (#3s)},#1}}
```

The `description` key will then need to be entered using the optional argument.

For example, if your document originally had the following:

```
\newacronym{SVM}{Support Vector Machine}{description=Statistical
pattern recognition technique}
```

Then you would need to redefine `\newacronym` as shown above, and change the above to:

```
\newacronym[description=Statistical pattern recognition
technique]{svm}{SVM}{Support Vector Machine}
```

You will also need to replace all occurrences of `\SVM` with `\gls{svm}`. Alternatively, you can define `\SVM`:

```
\newcommand{\SVM}{\gls{svm}}
```

If you have used `\useacronym` instead of `\`<*acr-name*>, then you will need to replace all occurrences of

`\useacronym[`<*insert*>`]{`<*acr-name*>`}`

glossary

with

`\gls{`<*label*>`}[`<*insert*>`]`

glossaries

Note that the starred versions of `\useacronym` and `\`<*acr-name*> (which make the first letter uppercase) should be replaced with `\Gls{`<*label*>`}`.

6.1 `\acrln` and `\acrsh`

In the glossary package, it is possible to produce the long and short forms of an acronym without adding an entry to the glossary using `\acrln` and `\acrsh`. With the glossaries package, you can replace

`\acrsh{`<*acr-name*>`}`

glossary

with

\glsentrytext{<*label*>}

glossaries

However the glossaries package provides no direct equivalent of `\acrln`. Instead it provides `\glsentryfirst` which produces the "first use" text without adding an entry to the glossary, or affecting the first use flag.

Alternatively, if you haven't changed the definition of `\newacronym`, the long form by itself can be obtained using `\glsentrydesc`, since by default the `description` key is set to the long form (but read the notes in the glossaries manual regarding the `sanitize` package option).

If you want to be able to set a description, and be able to independently access the description, the long form and the short form, you can redefine `\newacronym` so that it uses the `symbol` key and use `\defglsdisplayfirst` so that it uses this value on first use, as follows:

```
\renewcommand{\newacronym}[4][]{\newglossaryentry{#2}{%
name={#4 (#3)},
text={#3},
first={#4},
symbol={#3},
}}
\defglsdisplayfirst{\acronymtype}{##1 (##3)##4}
```

As before, you can access the short name via `\glsentrytext{`<*label*>`}`, but you can now access just the long form via `\glsentryfirst{`<*label*>`}` and the description via `\glsentrydesc{`<*label*>`}`.

See section 5.9 ("Using glossary entries in the text") of the glossaries manual for further details of how to use these commands.

6.2 `\ifacronymfirstuse`

The glossary package command

\ifacronymfirstuse{<*acr-name*>}{<*text1*>}{<*text2*>}

glossary

can be replaced by the glossaries command:

\ifglsused{<*label*>}{<*text2*>}{<*text1*>}

glossaries

Note that `\ifglsused` evaluates the opposite condition to that of `\ifacronymfirstuse` which is why the last two arguments have been reversed.

6.3 `\resetacronym` and `\unsetacronym`

The glossary package allows you to reset and unset the acronym flag which is used to determine whether the acronym has been used in the document. The

glossaries package also provides a means to do this on either a local or global level. To reset an acronym, you will need to replace:

```
\resetacronym{<acr-name>}
```

glossary

with either

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

glossaries

or

```
\glslocalreset{<label>}
```

glossaries

To unset an acronym, you will need to replace:

```
\unsetacronym{<acr-name>}
```

glossary

with either

```
\glsunset{<label>}
```

glossaries

or

```
\glslocalunset{<label>}
```

glossaries

7 Displaying the glossary

The glossary package provides the command `\printglossary` (or `\print`<*type*> for other glossary types) which can be used to print individual glossaries. The glossaries package provides the command `\printglossaries` which will print all the glossaries which have been defined, or `\printglossary[`<*options*>`]` to print individual glossaries. So if you just have `\printglossary`, then you can leave it as it is, but if you have, say:

```
\printglossary
\printglossary[acronym]
```

or

```
\printglossary
\printacronym
```

then you will need to either replace this with either

```
\printglossaries
```

or

```
\printglossary
\printglossary[type=\acronymtype]
```

The glossary package allows you to specify a short title (for the table of contents and page header) by defining a command of the form `\short`<*glossary-type*>`name`. The glossaries package doesn't do this, but instead provides the `toctitle` key which can be used in the optional argument to `\printglossary`. For example, if you have created a new glossary type called `notation`, and you had defined

```
\newcommand{\shortnotationname}{Notation}
```

then you would need to use the `toctitle` key:

```
\printglossary[type=notation,toctitle=Notation]
```

The glossaries package will ignore `\shortnotationname`, so unless you have used it elsewhere in the document, you may as well remove the definition.

8 Using makeindex

If you convert your document from using the glossary package to the glossaries package, you will need to delete any of the additional files, such as the `.glo` file, that were created by the glossary package, as the glossaries package uses a different `makeindex` style file. Remember also, that if you used the `makeglos` Perl script, you will need to use the `makeglossaries` Perl script instead.