

The makeglos package

Thomas Henlich (thenlich at arcort dot de)

March 17, 2005

Contents

Glossary	1
1 Introduction	1
2 Features	2
3 Commands	2
4 Notes	2

Glossary

Now follows a very short glossary:

glossary A list of words with explanations of these words

index processor a software product to prepare an index for inclusion into a \LaTeX document

xindy an index processor, can also be used to process glossaries; *see* index processor; *see also* glossary

1 Introduction

A glossary is a list of words (concepts, terms) together with their explanations. In a glossary you should explain those terms which are necessary for the understanding of your document, but which some of your readers might not know. The advantages of using a glossary vs. writing definitions in the middle of a sentence are:

- improved text flow: the explanations don't clutter up the logical structure of the text
- reader-friendliness: people who already know the meaning of terms (e. g. people who come from the same professional background as you) don't need to read their definitions. That helps them understand the text faster.

Very often a glossary is sorted alphabetically. An example for a glossary in a technical or scientific document is a list of mathematical symbols together with their meanings.

`makeglos` is a \LaTeX package to include a glossary into a document. The glossary must be prepared by an external program, like `xindy` or `makeindex`.

2 Features

- Can be easily configured: various aspects of the appearance of the glossary can be changed by a simple `\renewcommand`.
- Multi-language capability: certain keywords, like the name of the glossary, can be made language-dependent.
- Flexible: works equally well for document classes with a book- or report-like structure (divided into `\chapters`) and for article-like documents, like the one you are reading now (based on `\sections`)
- Equivalent: `makeglos` is to glossaries what `makeidx` is to indices.

3 Commands

<code>\printglossary</code>	Inserts the glossary file (<code>*.gls</code>) here.
<code>\glossaryname</code>	The name of the glossary. Defaults to “Glossary”.
<code>\glossaryintro</code>	Something to print before the actual glossary. Defaults to empty.
<code>\gsee</code>	The cross-reference introduction phrase. Defaults to <code>\emph{\seename} #1</code> .
<code>\galso</code>	The secondary cross-reference introduction phrase. Defaults to <code>\emph{\alsoname} #1</code> .
<code>\seename</code>	The cross-reference introduction phrase. Defaults to “see”. Redefined by <code>babel</code> package.
<code>\alsoname</code>	The secondary cross-reference introduction phrase. Defaults to “see also”. Redefined by <code>babel</code> package.

4 Notes

The glossary environment is by default defined to be a chapter (or section) which does not appear in the table of contents. If you want the glossary to appear there (just like I did when I wrote this), just add

```
\addcontentsline{toc}{section}{\glossaryname}.
```

The glossary’s environment definition for document classes in which the `\chapter` command is undefined (article-like documents) is

```
\newenvironment{theglossary}%  
{\section*{\glossaryname}\glossaryintro%  
\begin{description}}%  
\end{description}}%
```

The glossary's environment definition for document classes in which the `\chapter` command is defined (report-/book-like documents) is

```
\newenvironment{theglossary}%  
{\chapter*{\glossaryname}\glossaryintro%  
\begin{description}}%  
\end{description}}%
```

If you want to change this, `\renewenvironment` is your friend!

The glossary for this document was prepared with `xindy`, version 2.0d:
`xindy -o makeglos.gls -f tex2xindy makeglos.xdy makeglos.glo`