

The miunmisc package*

Daniel Bosk
daniel.bosk@miun.se

October 8, 2015

Contents

| | | |
|----------|----------------------------------|----------|
| 1 | Introduction | 1 |
| 2 | Usage | 1 |
| 3 | Implementation | 2 |
| 3.1 | varioref and prettyref | 2 |
| 3.2 | listings | 5 |
| 3.3 | natbib | 9 |
| 3.4 | algorithm | 10 |
| 3.5 | beamer | 12 |
| 3.6 | nomencl | 12 |
| 3.7 | Other options | 13 |
| 3.8 | Licensing macros | 13 |

1 Introduction

The miunmisc package provides some default settings for various packages, this solution is more general than to change these settings in all `miun*` document classes. This way the `miunmisc` settings can also be used with other classes than the MiUn specific classes.

The package was initially created by Daniel Bosk and Lennart Franked, but also contains contributions from Martin Kjellqvist.

2 Usage

To use the `miunmisc` package, simply load the package using the `\usepackage` command with the options which are applicable for the document in question. One thing to note is that `miunmisc` must be loaded *last*, or at least after all packages which are to be set up by `miunmisc`.

*This document corresponds to `miunmisc` v1.7, dated 2013/05/05.

3 Implementation

The `miunmisc` package uses the `translator` package for translations of section names and names of floats. Currently, and probably forever, only English and Swedish are supported.

```
1 <*package>
2 \@ifpackageloaded{translator}{%
3   \PassOptionsToPackage{english,swedish}{translator}
4   \usedictionary{miunmisc}
5 }{%
6   \RequirePackage[english,swedish]{translator}
7   \usedictionary{miunmisc}
8 }
9 </package>
```

Hence, two dictionaries are provided, one with English and another with Swedish translations.

```
10 <*dicteng>
11 \ProvidesDictionary{miunmisc}{English}
12 </dicteng>
13 <*dictswe>
14 \ProvidesDictionary{miunmisc}{Swedish}
15 </dictswe>
```

`miunmisc` uses options passed to it via the `\usepackage` command in the preamble of the document. When the package is loaded and the options parsed the specific settings are loaded.

3.1 `varioref` and `prettyref`

Two packages which are extensively used are `varioref` and `prettyref`.

`varioref` The option `varioref` sets up the translation of all reference strings. If the `babel` package is loaded we use its `\addto` command to be able to correctly switch between languages.

```
16 <*package>
17 \DeclareOption{varioref}{%
18   \@ifpackageloaded{babel}{%
19     \addto{\extrasswedish}{%
20       \renewcommand{\reftextbefore}{\translate{textbefore}}%
21       \renewcommand{\reftextfacebefore}{\translate{textfacebefore}}%
22       \renewcommand{\reftextafter}{\translate{textafter}}%
23       \renewcommand{\reftextfaceafter}{\translate{textfaceafter}}%
24       \renewcommand{\reftextfaraway}[1]{\translate{textfaraway} \pageref{#1}}%
25     }
26     \addto{\extrasenglish}{%
27       \renewcommand{\reftextbefore}{\translate{textbefore}}%
28       \renewcommand{\reftextfacebefore}{\translate{textfacebefore}}%
29       \renewcommand{\reftextafter}{\translate{textafter}}%
30       \renewcommand{\reftextfaceafter}{\translate{textfaceafter}}%
31       \renewcommand{\reftextfaraway}[1]{\translate{textfaraway} \pageref{#1}}%
32     }
33   }{%
34     \renewcommand{\reftextbefore}{\translate{textbefore}}
```

```

35 \renewcommand{\reftextfacebefore}{\translate{textfacebefore}}
36 \renewcommand{\reftextafter}{\translate{textafter}}
37 \renewcommand{\reftextfaceafter}{\translate{textfaceafter}}
38 \renewcommand{\reftextfaraway}[1]{\translate{textfaraway} \pageref{#1}}%
39 }
40 }
41 \end{package}

```

The translations used in this code are provided in each dictionary.

```

42 \begin{dicteng}
43 \providetranslation{textbefore}{on the previous page}
44 \providetranslation{textfacebefore}{on the facing page}
45 \providetranslation{textafter}{on the next page}
46 \providetranslation{textfaceafter}{on the facing page}
47 \providetranslation{textfaraway}{on page}
48 \end{dicteng}
49 \begin{dictswe}
50 \providetranslation{textbefore}{p{\aa} f{"o}reg{\aa}ende sida}
51 \providetranslation{textfacebefore}{p{\aa} f{"o}reg{\aa}ende sida}
52 \providetranslation{textafter}{p{\aa} n{"a}sta sida}
53 \providetranslation{textfaceafter}{p{\aa} n{"a}sta sida}
54 \providetranslation{textfaraway}{p{\aa} sidan}
55 \end{dictswe}

```

prettyref The `prettyref` option does the same thing as `varioref`, except for the `prettyref` package instead. However, it uses the `varioref` package. If the `varioref` package is not loaded, it uses the standard `\ref` command.

```

56 \begin{package}
57 \DeclareOption{prettyref}{%
58   \@ifpackageloaded{varioref}{\relax}{%
59     \let\vref\ref
60   }
61 \end{package}

```

It also uses the `\eqref` command from the `amsmath` package if loaded. Otherwise, it defines its own version of the command.

```

62 \begin{package}
63   \@ifpackageloaded{amsmath}{\relax}{%
64     \DeclareRobustCommand{\eqref}[1]{%
65       (\ref{#1})
66     }
67   }
68 \end{package}

```

As with the `varioref` option, the `prettyref` option uses `babel` functionality if that package is loaded.

```

69 \begin{package}
70   \@ifpackageloaded{babel}{%
71     \addto{\extrasswedish}{%
72       \newreformat{prt}{\translate{part} \ref{#1}}%
73       \newreformat{ch}{\translate{chapter} \ref{#1}}%
74       \newreformat{sec}{\translate{section} \ref{#1}}%
75       \newreformat{app}{\translate{appendix} \ref{#1}}%
76       \newreformat{q}{\translate{question} \vref{#1}}%
77       \newreformat{qp}{\translate{part} \ref{#1}}%

```

```

78     \newreformat{xrc}{\translate{exercise} \vref{#1}}%
79     \newreformat{eq}{\translate{equation} \eqref{#1}}%
80     \newreformat{tab}{\translate{table} \vref{#1}}%
81     \newreformat{tbl}{\translate{table} \vref{#1}}%
82     \newreformat{fig}{\translate{figure} \vref{#1}}%
83     \newreformat{alg}{\translate{algorithm} \vref{#1}}%
84     \newreformat{lst}{\translate{listing} \vref{#1}}%
85     \newreformat{ln}{\translate{line} \ref{#1}}%
86 }
87 \addto{\extrasenglish}{%
88     \newreformat{prt}{\translate{Part} \ref{#1}}%
89     \newreformat{ch}{\translate{Chapter} \ref{#1}}%
90     \newreformat{sec}{\translate{Section} \ref{#1}}%
91     \newreformat{app}{\translate{Appendix} \ref{#1}}%
92     \newreformat{q}{\translate{question} \vref{#1}}%
93     \newreformat{qp}{\translate{part} \ref{#1}}%
94     \newreformat{xrc}{\translate{exercise} \vref{#1}}%
95     \newreformat{eq}{\translate{equation} \eqref{#1}}%
96     \newreformat{tab}{\translate{Table} \vref{#1}}%
97     \newreformat{tbl}{\translate{Table} \vref{#1}}%
98     \newreformat{fig}{\translate{Figure} \vref{#1}}%
99     \newreformat{alg}{\translate{Algorithm} \vref{#1}}%
100    \newreformat{lst}{\translate{Listing} \vref{#1}}%
101    \newreformat{ln}{\translate{line} \ref{#1}}%
102 }
103 }{%
104     \newreformat{prt}{\translate{Part} \ref{#1}}
105     \newreformat{ch}{\translate{Chapter} \ref{#1}}
106     \newreformat{sec}{\translate{Section} \ref{#1}}
107     \newreformat{app}{\translate{Appendix} \ref{#1}}
108     \newreformat{q}{\translate{question} \vref{#1}}
109     \newreformat{qp}{\translate{part} \ref{#1}}
110     \newreformat{xrc}{\translate{exercise} \vref{#1}}
111     \newreformat{eq}{\translate{equation} \eqref{#1}}
112     \newreformat{tab}{\translate{Table} \vref{#1}}
113     \newreformat{tbl}{\translate{Table} \vref{#1}}
114     \newreformat{fig}{\translate{Figure} \vref{#1}}
115     \newreformat{alg}{\translate{Algorithm} \vref{#1}}
116     \newreformat{lst}{\translate{Listing} \vref{#1}}
117     \newreformat{ln}{\translate{line} \ref{#1}}
118 }
119 }
120 \end{package}

```

The translations used herein are again provided in the dictionaries.

```

121 \begin{dicteng}
122 \providetranslation{Part}{Part}
123 \providetranslation{part}{part}
124 \providetranslation{Chapter}{Chapter}
125 \providetranslation{chapter}{chapter}
126 \providetranslation{Section}{Section}
127 \providetranslation{section}{section}
128 \providetranslation{Appendix}{Appendix}
129 \providetranslation{appendix}{appendix}
130 \providetranslation{Question}{Question}

```

```

131 \providetranslation{question}{question}
132 \providetranslation{Equation}{Equation}
133 \providetranslation{equation}{equation}
134 \providetranslation{Table}{Table}
135 \providetranslation{table}{table}
136 \providetranslation{Figure}{Figure}
137 \providetranslation{figure}{figure}
138 \providetranslation{Algorithm}{Algorithm}
139 \providetranslation{algorithm}{algorithm}
140 \providetranslation{Exercise}{Exercise}
141 \providetranslation{exercise}{exercise}
142 \providetranslation{unsrtnat}{unsrtnat}
143 \providetranslation{Listing}{Listing}
144 \providetranslation{listing}{listing}
145 \providetranslation{Line}{Line}
146 \providetranslation{line}{line}
147 </dicteng>
148 <*dictswe>
149 \providetranslation{Part}{Del}
150 \providetranslation{part}{del}
151 \providetranslation{Chapter}{Kapitel}
152 \providetranslation{chapter}{kapitel}
153 \providetranslation{Section}{Avsnitt}
154 \providetranslation{section}{avsnitt}
155 \providetranslation{Appendix}{Bilaga}
156 \providetranslation{appendix}{bilaga}
157 \providetranslation{Question}{Fr{\aa}ga}
158 \providetranslation{question}{fr{\aa}ga}
159 \providetranslation{Equation}{Ekvation}
160 \providetranslation{equation}{ekvation}
161 \providetranslation{Table}{Tabell}
162 \providetranslation{table}{tabell}
163 \providetranslation{Figure}{Figur}
164 \providetranslation{figure}{figur}
165 \providetranslation{Algorithm}{Algoritm}
166 \providetranslation{algorithm}{algoritm}
167 \providetranslation{Exercise}{Uppgift}
168 \providetranslation{exercise}{uppgift}
169 \providetranslation{unsrtnat}{swepnat}
170 \providetranslation{Listing}{Listning}
171 \providetranslation{listing}{listning}
172 \providetranslation{Line}{Rad}
173 \providetranslation{line}{rad}
174 </dictswe>

```

3.2 listings

`listings` The `listings` option sets up, as the name suggests, the `listings` package. If the package `color` is loaded it defines some basic colours used by the `listings`.

```

175 <*package>
176 \DeclareOption{listings}{%
177   \@ifpackageloaded{color}{%
178     \definecolor{termbkg}{gray}{0.90}%

```

```

179 \definecolor{textbkg}{gray}{0.97}%
180 \definecolor{commentcol}{rgb}{0,0.5,0}%
181 }{% color not loaded
182 \relax
183 }
184 </package>

```

The next thing this option does is to make sure the swedish letters å, ä and ö works properly in code listings.

```

185 <*package>
186 \lstset{literatename={
187   {Ö}{{\ "O}}1
188   {Å}{{\ "A}}1
189   {å}{{\ "aa}}1
190   {ä}{{\ "a}}1
191   {ö}{{\ "o}}1
192 }
193 }
194 </package>

```

Next we set up the default listing style. The text and line numbers should be typeset using small fonts. We also want to show the spaces within strings, but not in the code. Lines will be broken at whitespaces, not in the middle of words. The caption should be at the bottom.

```

195 <*package>
196 \lstset{%
197   basicstyle=\small,
198   numberstyle=\small,
199   numbers=left,
200   stepnumber=1,
201   numbersep=5pt,
202   showspaces=false,
203   showstringspaces=true,
204   showtabs=false,
205   frame=TB,
206   tabsize=2,
207   captionpos=b,
208   breaklines=true,
209   breakatwhitespace=true,
210 }
211 </package>

```

When typesetting output from a terminal we do not want any syntax highlighting. To accomplish this we use a language consisting of no keywords, this language is defined next.

```

212 <*package>
213 \lstdefinlanguage{none}{
214   keywords={ }
215 }
216 </package>

```

We provide predefined styles for typesetting different types of listings. One for terminal output, one for text files, and a last one for source code. If the color package is loaded we use different background colors for each of the styles.

```

217 <*package>

```

```

218 \@ifpackageloaded{color}{
219   \lstdefinestyle{term}{
220     language=none,
221     numbers=left,
222     frame=single,
223     basicstyle=\ttfamily\small,
224     backgroundcolor=\color{termbkg},
225   }
226 \lstdefinestyle{conf}{
227   language=none,
228   numbers=none,
229   frame=single,
230   basicstyle=\ttfamily\small,
231   backgroundcolor=\color{textbkg},
232 }
233 \lstdefinestyle{text}{
234   language=none,
235   numbers=none,
236   frame=single,
237   basicstyle=\ttfamily\small,
238   backgroundcolor=\color{textbkg},
239 }
240 \lstdefinestyle{code}{
241   numbers=left,
242   frame=single,
243   basicstyle=\ttfamily\small,
244   backgroundcolor=\color{white},
245 }
246 }{
247   \lstdefinestyle{term}{
248     language=none,
249     numbers=left,
250     frame=single,
251     basicstyle=\ttfamily\small,
252   }
253 \lstdefinestyle{conf}{
254   language=none,
255   numbers=none,
256   frame=single,
257   basicstyle=\ttfamily\small,
258 }
259 \lstdefinestyle{text}{
260   language=none,
261   numbers=none,
262   frame=single,
263   basicstyle=\ttfamily\small,
264 }
265 \lstdefinestyle{code}{
266   numbers=left,
267   frame=single,
268   basicstyle=\ttfamily\small,
269 }
270 }
271 \end{package}

```

The default listings style is set to plain text.

```
272 <*package>
273 \lstset{style=text}
274 </package>
```

Next we use these styles to create environments using them, to make inline listings easier to handle.

src The first is the **source** environment. The environment **src** is just a shorthand for **source**. You use them simply by issuing `\begin{source}[\lstlistings args]` and the corresponding `\end{source}` with all text between them read verbatim as source code.

```
275 <*package>
276 \lstnewenvironment{src}[1] [] {%
277   \lstset{style=code,#1}
278 }{%
279   \lstset{style=text}
280 }
281 \lstnewenvironment{source}[1] [] {%
282   \lstset{style=code,#1}
283 }{%
284   \lstset{style=text}
285 }
286 </package>
```

The optional arguments are the same as the optional arguments otherwise passed to the `lstlistings` environment. Typically optional arguments are used to specify a language or setting the listing as a float.

textfile The same applies to the final three environments, **textfile**, **config** and **terminal**, with the exception that it is unlikely that one wish to specify a language for these listings.

```
287 <*package>
288 \lstnewenvironment{textfile}[1] [] {%
289   \lstset{style=text,#1}
290 }{%
291   \lstset{style=text}
292 }
293 \lstnewenvironment{config}[1] [] {%
294 \lstset{style=conf,#1}
295 }{%
296 \lstset{style=conf}
297 }
298 \lstnewenvironment{terminal}[1] [] {%
299   \lstset{style=term,#1}
300 }{%
301   \lstset{style=text}
302 }
303 </package>
```

We also provide some shorter commands for inline writing.

\code First we have the command `\code[\lstlisting args]{(the code)}` which will typeset code inline using the `\lstinline` command.


```

304 (*package)
305 \newcommand{\code}[2][style=code]{\lstinline[#1]‘#2’}
306 \end{package}

```

Then we provide the command `\term`[*listing args*]{*terminal commands*} which will typeset terminal commands.

```

307 (*package)
308 \newcommand{\term}[2][style=term]{\lstinline[#1]‘#2’}
309 \end{package}

```

Finally, we use translations for the names of the floats and the list of listings.

```

310 (*package)
311 \renewcommand{\lstlistlistingname}{\translate{Listings}}
312 \renewcommand{\lstlistingname}{\translate{Listing}}
313 }
314 \end{package}

```

And then the actual dictionary entries.

```

315 (*dicteng)
316 \providetranslation{Listings}{Listings}
317 \providetranslation{listings}{listings}
318 \end{dicteng}
319 (*dictswe)
320 \providetranslation{Listings}{Listningar}
321 \providetranslation{listings}{listningar}
322 \end{dictswe}

```

3.3 natbib

`natbib` The `natbib` option sets up the `natbib` package. The current setup is to use the `unsrtnat` bibliography style, and if `babel` is set to use Swedish we use the Swedish version, namely `sweplnat`.

```

323 (*package)
324 \DeclareOption{natbib}{%
325   \@ifpackageloaded{babel}{%
326     \iflanguage{swedish}{%
327       \bibliographystyle{sweplnat}%
328     }{
329       \bibliographystyle{unsrtnat}%
330     }
331   }{%
332     \relax%
333   }
334 \end{package}

```

We also want to use a Vancouver-based citation style for references, i.e. numbers with brackets.

```

335 (*package)
336 \setcitestyle{numbers,square}
337 }
338 \end{package}

```

Unfortunately, the IEEE and alpha bibliography styles are not supported by `natbib`, these have to be rewritten to support exporting the author name and the year of publication. When this is done the IEEE will be the default set by `mionmisc`.

3.4 algorithm

`algorithm` The `algorithm` option sets up the packages `algorithm` and `algpseudocode`. What it first does is to make sure the pseudocode floats are typeset similar to the standard floats, e.g. figures and tables.

```
339 <*package>
340 \DeclareOption{algorithm}{%
341   \ifx\ALG@floatstyle\undefined\relax\else
342     \renewcommand{\ALG@floatstyle}{plain}
343     \floatstyle{plain}
344     \restylefloat{algorithm}
345   \fi
346 </package>
```

What it does next is to load all translations. As previously, if the `babel` package is loaded we use its `\addto` functionality. This allows us to write pretty Swedish pseudocode as well.

```
347 <*package>
348   \@ifpackageloaded{babel}{%
349     \ifx\floatname\undefined\relax\else%
350       \addto{\extrasswedish}{\floatname{algorithm}{\translate{Algorithm}}}
351       \addto{\extrasenglish}{\floatname{algorithm}{\translate{Algorithm}}}
352     \fi
353   \@ifpackageloaded{algpseudocode}{%
354     \addto{\extrasswedish}{%
355       \algnewcommand\algorithmicfunction{\textbf{\translate{function}}}%
356       \algnewcommand\algorithmicprocedure{\textbf{\translate{procedure}}}%
357       \algnewcommand\algorithmicwhile{\textbf{\translate{while}}}%
358       \algnewcommand\algorithmicdo{\textbf{\translate{do}}}%
359       \algnewcommand\algorithmicend{\textbf{\translate{end}}}%
360       \algnewcommand\algorithmicforall{\textbf{\translate{for all}}}%
361       \algnewcommand\algorithmicfor{\textbf{\translate{for}}}%
362       \algnewcommand\algorithmicrepeat{\textbf{\translate{repeat}}}%
363       \algnewcommand\algorithmicuntil{\textbf{\translate{until}}}%
364       \algnewcommand\algorithmicloop{\textbf{\translate{loop}}}%
365       \algnewcommand\algorithmicif{\textbf{\translate{if}}}%
366       \algnewcommand\algorithmicthen{\textbf{\translate{then}}}%
367       \algnewcommand\algorithmicelse{\textbf{\translate{else}}}%
368       \algnewcommand\algorithmicrequire{\textbf{\translate{require}}}%
369       \algnewcommand\algorithmicensure{\textbf{\translate{ensure}}}%
370       \algnewcommand\algorithmicreturn{\textbf{\translate{return}}}%
371     }{\relax}
372     \addto{\extrasenglish}{%
373       \algnewcommand\algorithmicfunction{\textbf{\translate{function}}}%
374       \algnewcommand\algorithmicprocedure{\textbf{\translate{procedure}}}%
375       \algnewcommand\algorithmicwhile{\textbf{\translate{while}}}%
376       \algnewcommand\algorithmicdo{\textbf{\translate{do}}}%
377       \algnewcommand\algorithmicend{\textbf{\translate{end}}}%
378       \algnewcommand\algorithmicforall{\textbf{\translate{for all}}}%
379       \algnewcommand\algorithmicfor{\textbf{\translate{for}}}%
380       \algnewcommand\algorithmicrepeat{\textbf{\translate{repeat}}}%
381       \algnewcommand\algorithmicuntil{\textbf{\translate{until}}}%
382       \algnewcommand\algorithmicloop{\textbf{\translate{loop}}}%
383       \algnewcommand\algorithmicif{\textbf{\translate{if}}}%

```

```

384     \algnewcommand\algorithmicthen{\textbf{\translate{then}}}%
385     \algnewcommand\algorithmicelse{\textbf{\translate{else}}}%
386     \algnewcommand\algorithmicrequire{\textbf{\translate{require}}}%
387     \algnewcommand\algorithmicensure{\textbf{\translate{ensure}}}%
388     \algnewcommand\algorithmicreturn{\textbf{\translate{return}}}%
389     }\relax}
390 }{\relax}
391 }{%
392 \ifpackageloaded{algpseudocode}{%
393     \algnewcommand\algorithmicfunction{\textbf{\translate{function}}}%
394     \algnewcommand\algorithmicprocedure{\textbf{\translate{procedure}}}%
395     \algnewcommand\algorithmicwhile{\textbf{\translate{while}}}%
396     \algnewcommand\algorithmicdo{\textbf{\translate{do}}}%
397     \algnewcommand\algorithmicend{\textbf{\translate{end}}}%
398     \algnewcommand\algorithmicforall{\textbf{\translate{for all}}}%
399     \algnewcommand\algorithmicfor{\textbf{\translate{for}}}%
400     \algnewcommand\algorithmicrepeat{\textbf{\translate{repeat}}}%
401     \algnewcommand\algorithmicuntil{\textbf{\translate{until}}}%
402     \algnewcommand\algorithmicloop{\textbf{\translate{loop}}}%
403     \algnewcommand\algorithmicif{\textbf{\translate{if}}}%
404     \algnewcommand\algorithmicthen{\textbf{\translate{then}}}%
405     \algnewcommand\algorithmicelse{\textbf{\translate{else}}}%
406     \algnewcommand\algorithmicrequire{\textbf{\translate{require}}}%
407     \algnewcommand\algorithmicensure{\textbf{\translate{ensure}}}%
408     \algnewcommand\algorithmicreturn{\textbf{\translate{return}}}%
409     }\relax}
410 }
411 }
412 \end{package}

```

And then we provide the dictionary entries.

```

413 (*dicteng)
414 \providetranslation{function}{function}
415 \providetranslation{procedure}{procedure}
416 \providetranslation{while}{while}
417 \providetranslation{do}{do}
418 \providetranslation{end}{end}
419 \providetranslation{for all}{for all}
420 \providetranslation{for}{for}
421 \providetranslation{repeat}{repeat}
422 \providetranslation{until}{until}
423 \providetranslation{loop}{loop}
424 \providetranslation{if}{if}
425 \providetranslation{then}{then}
426 \providetranslation{else}{else}
427 \providetranslation{require}{input}
428 \providetranslation{ensure}{output}
429 \providetranslation{return}{return}
430 \end{dicteng}
431 (*dictswe)
432 \providetranslation{function}{funktion}
433 \providetranslation{procedure}{procedur}
434 \providetranslation{while}{medan}
435 \providetranslation{do}{genomf{"o}r}
436 \providetranslation{end}{slut}

```

```

437 \providetranslation{for all}{f{"o}r alla}
438 \providetranslation{for}{f{"o}r}
439 \providetranslation{repeat}{repetera}
440 \providetranslation{until}{tills}
441 \providetranslation{loop}{iterera}
442 \providetranslation{if}{om}
443 \providetranslation{then}{d{\aa}}
444 \providetranslation{else}{annars}
445 \providetranslation{require}{indata}
446 \providetranslation{ensure}{utdata}
447 \providetranslation{return}{returnera}
448 \dictswe

```

3.5 beamer

beamer The `beamer` option adds some features to the `beamer` document class. It adds translations for some of `beamer`'s blocks, e.g. `theorem` and `example`.

```

449 \package
450 \DeclareOption{beamer}{%
451   \providetranslation[to=swedish]{Theorem}{Sats}
452   \providetranslation[to=swedish]{Corollary}{Korollarium}
453   \providetranslation[to=swedish]{Axiom}{Axiom}
454   \providetranslation[to=swedish]{Example}{Exempel}
455   \providetranslation[to=swedish]{Algorithm}{Algoritm}
456 }

```

It also adds the environments `axiom` and `algorithm` which does similar to the `theorem` environment already in `beamer`. Both these environments currently takes one mandatory argument which serves as a title.

```

457 \package
458 \newenvironment{axiom}[1]{\begin{block}{\translate{Axiom} (#1)}}{\end{block}}
459 \@ifpackageloaded{algorithm}{\relax}{%
460   \newenvironment{algorithm}[1]{\begin{block}{\translate{Algorithm}
461     (#1)}}{\end{block}}
462 }
463 }
464 \package

```

3.6 nomencl

nomencl The `nomencl` option changes some of the defaults for the `nomencl` package.

```

465 \package
466 \DeclareOption{nomencl}{%
467   \renewcommand{\nomname}{\translate{Nomenclature}}
468   \renewcommand{\eqdeclaration}[1]{, \translate{see equation}\nobreakspace#1}
469   \renewcommand{\pagedeclaration}[1]{, \translate{see page}\nobreakspace#1}
470 }
471 \package

```

It also adds support for Swedish.

```

472 \dictswe
473 \providetranslation{Nomenclature}{Nomenklatur}

```

```

474 \providetranslation{see equation}{se ekvation}
475 \providetranslation{see page}{se sidan}
476 </dictswe>
477 <*dicteng>
478 \providetranslation{Nomenclature}{Nomenclature}
479 \providetranslation{see equation}{see equation}
480 \providetranslation{see page}{see page}
481 </dicteng>

```

3.7 Other options

parskip The option `parskip` is the "Lennart option". He once argued that paragraphs should be indicated by a small vertical space skip instead of the first line of the paragraph being indented. Of course, he is not wrong to argue this way, it is approved in standard writing guides for Swedish. However, from a typesetting perspective it is flawed. Thus, with the manual for the memoir document class as a reference, I hereby declare that *use of this option is strongly discouraged!*

```

482 (*package)
483 \DeclareOption{parskip}{% The Lennart option, do not use! [texdoc memman]
484   \setlength{\parskip}{5pt}
485   \setlength{\parindent}{0pt}
486 }
487 </package>

```

color The option `color` sets up the colours specified by the Mid Sweden University graphical profile.

```

488 (*package)
489 \DeclareOption{color}{%
490   \definecolor{miunyellow}{cmyk}{0,0.10,1.0,0}
491   \definecolor{miunblue}{cmyk}{1.0,0.34,0,0.02}
492   \definecolor{miunblack}{cmyk}{0,0,0,1.0}
493   \definecolor{miungray}{cmyk}{0,0,0,0.11}
494   \definecolor{miundark}{cmyk}{0.11,0.01,0,0.69}
495   \definecolor{miunwhite}{cmyk}{0,0,0,0}
496 }
497 </package>

```

And finally, we have the options processed by the package.

```

498 (*package)
499 \ProcessOptions\relax
500 </package>

```

3.8 Licensing macros

The `miunmisc` package also provides some useful macros. Currently it adds macros to easily typeset various licenses. Currently Creative Commons Attribution-ShareAlike in Swedish and English are supported.

\CCBYSA The `\CCBYSA` macro inserts a text saying that the work is licensed under the Creative Commons Attribution-ShareAlike license.

```

501 (*package)

```

```

502 \DeclareRobustCommand{\CCBYSA}{%
503   \translate{This work is published under the}%
504   \translate{CC BY-SA}.
505 }
506 \</package>

  It uses babel and translator to determine whether the English or Swedish version
  of the license should be used.

507 \<dicteng>
508 \providetranslation{This work is published under the}{%
509   This work is published under the
510 }
511 \providetranslation{CC BY-SA}{%
512   Creative Commons Attribution-ShareAlike 3.0 Unported license, to view a copy
513   of this license please visit
514   \url{http://creativecommons.org/licenses/by-sa/3.0/}
515 }
516 \</dicteng>
517 \<dictswe>
518 \providetranslation{This work is published under the}{%
519   Detta verk är publicerat under licensen
520 }
521 \providetranslation{CC BY-SA}{%
522   Creative Commons Erkännande-DelaLika 2.5 Sverige (CC BY-SA 2.5 SE), för att
523   se en sammanfattning och fullständig kopia av licenstexten besök URL
524   \url{http://creativecommons.org/licenses/by-sa/2.5/se/}
525 }
526 \</dictswe>

```