

Babel, how to enjoy writing in different languages

Enrico Gregorio

Email Enrico.Gregorio@univr.it
Address Dipartimento di Informatica
Università di Verona
Italy

1 Introduction

Anybody who needs to write in languages different from English should know about babel. At the beginning of L^AT_EX tags like ‘Chapter’ or ‘Table of Contents’ were hardwired in the format and changing them required hunting for those strings in the file `lplain.tex` and doing things like

```
\makeatletter  
\renewcommand{\@chapapp}{Capitolo}  
\makeatother
```

just to change the tag for Italian. In those days, customizing L^AT_EX for a non-English language was a nightmare, not to mention the fact that you could use only one language at a time, so canning, say, `itlplain.fmt` from a modified source of L^AT_EX which loaded Italian hyphenation patterns.

Then it came Johannes Braams with his babel which acted on the innards of L^AT_EX by changing the tags to suit many European languages. It was still necessary to build different formats for different languages, this was finally overcome by the introduction of T_EX3 with its multilingual capabilities.

And then it came Thomas Esser with teT_EX and other distributions for different operating systems which made easier to define a unique format with many hyphenation patterns available.

Later again L^AT_EX2_ε allowed for simplified syntax in the loading of packages, with the possibility of passing them *options*. Now Spaniards can say

```
\documentclass[a4paper]{book}  
\usepackage[spanish]{babel}
```

and their first chapter will be named, accordingly, *Capítulo*.

In the meantime babel has been incorporated in every official distribution of L^AT_EX, so that everyone can benefit from it.

We won't go into the details, focusing instead on the use of babel, but a word about how this package accomplishes its tasks can be useful. All fixed tags are put into commands; for example, the standard L^AT_EX class book defines

```
\newcommand{\chaptername}{Chapter}
```

and uses `\chaptername` in the definition of `\chapter`. Thus it is easy for babel to change: it is sufficient for it to issue the command

```
\renewcommand{\chaptername}{Capítulo}
```

when it loads the settings for Spanish. Well, actually babel doesn't do in precisely this way, but this is unimportant from a user's point of view. For users the good news is that babel changes the tags *automatically*.

2 Loading the package

The babel is loaded in the usual way

```
\usepackage[languages]{babel}
```

where *languages* denotes a comma separated list of language names as understood by babel; you find a list in Table 1. Just to be explicit, these are examples of correct usage:

```
\usepackage[italian]{babel}
```

```
\usepackage[frenchb,english]{babel}
```

```
\usepackage[greek,uppersorbian,ukrainian]{babel}
```

The rule is to declare all languages that will be needed in the document, keeping *last* the main one, that we desire to typeset the tags with. Some of the names are just synonyms of each other (portuges and portuguese, for example, or acadian and canadien; of course, canadian is not synonymous with canadien). Some of the names are there for historical reasons and mean just the same thing (this is the case of french, francais and frenchb).

One notable exception is english, which could mean either British English or American English, depending on the choices of the manager of the T_EX system

Table 1 List of babel languages

acadian	estonian	ngerman
afrikaans	finnish	norsk
albanian	francais	nynorsk
american	french	polish
australian	frenchb	polutonikogreek
austrian	galician	portuges
bahasa	german	portuguese
bahasai	germanb	romanian
basam	greek	russian
basque	hebrew	samin
brazil	hungarian	scottish
brazilian	icelandic	serbian
breton	indon	slovak
british	indonesian	slovene
bulgarian	interlingua	spanish
canadian	irish	swedish
canadien	italian	turkish
catalan	latin	ukrainian
croatian	lowersorbian	uppersorbian
czech	magyar	welsh
danish	malay	UKenglish
dutch	meyalu	USenglish
english	naustrian	
esperanto	newzealand	

you are using; usually it means the US variety. If you are unsure about that choice, you can specify UKenglish or USenglish as option to the package.

Another peculiarity worth noting is the presence of the pairs *german*, *ngerman* and *austrian*, *naustrian*. The *n*-versions refer to the *New Orthography* (*Neue Rechtschreibung*) of German. In Table 2 we find how ‘Chapter’ is written in some languages.

Table 2 The word ‘Chapter’ in some languages

Brazilian	Czech	Greek	Samin	Ukrainian	Welsh
Capítulo	Kapitola	Κεφάλαιο	Kapihttal	Розділ	Pennod

Some readers not accustomed with European countries will be astonished by the number of languages. They will be amazed on learning that at least as many are missing than are supported. There are not Lithuanian or Luxembourgish among the others; tens of local languages are not worth supporting, as their native speakers are so few. Luxembourg is one of the most important cities of the European Union, seat of the European Court of Justice; Lithuania has one of the best basketball national teams in the world. One of the four official languages of Switzerland does not appear in the list, Romansh.

Anyway, let’s return to the supported languages. If your document uses only one of them, say Portuguese, then the invocation should be

```
\usepackage[portuges]{babel}
```

You could also give *portuges* as option to `\documentclass`, so that you pass it also to all packages that understand it, for example *varioref*. No problem here, just start writing and all your tags will be in Portuguese: the *thebibliography* environment in an article will start a new section named *Referências*.

Now here it comes the difficult part: writing a document which requires at least two languages. Say a literature commentary where some citations in the original language are needed. Let’s say I’m Italian and I want to write a paper on the Czech writer Karel Čapek, the inventor of the word *robot* in the book “RUR, Rossumovi univerzální roboti” published in 1920.

My problem is that I don’t want to type my paper with a Unicode keyboard because input would be painful. I would have to choose between a Latin1 or a Latin2 keyboard and none of them is suited for both languages: while Latin2 has all the diacritics needed for Czech, it lacks all grave accents on vowels that I must use in Italian. Then I make a decision: Italian is the main language, so I choose a Latin1 keyboard and use *babel* facilities for typing Czech.

In Table 3 you find some text in the two languages, while in Table 4 is shown how to input the Czech text.

Table 3 An example of text in two languages. First the beginning of “RUR”, in Czech, then a short passage on the life and works of Karel Čapek, in Italian, with an explanation of the above passage.

Domin: (sedí u velikého amerického psacího stolu v otáčecím křesle. Na stole žárovka, telefon, těžítka, pořadač dopisů, atd., na stěně vlevo velké mapy s lodními a železničními liniemi, veliký kalendář, hodiny, jež ukazují něco málo před polednem; na stěně vpravo tištěné plakáty: “Nejlacinější práce: Rossumovi Roboti” “Tropičtí Roboti, nový vynález. Kus 150 d.” “Každý si kup svého Robota!” “Chcete zlevnit svoje výrobky? Objednejte si Rossumovy Roboty.” Dále jiné mapy, dopravní lodní řád, tabulka s telegrafickými záznamy kursů atd. V kontrastu k této výzdobě stěn je na zemi nádherný turecký koberec, vpravo kulatý stůl, pohovka, kožená klubovní křesla a knihovna, v níž místo knih stojí láhve s vínem a kořalkami. Vlevo pokladna. Vedle Dominova stolu psací stroj, na němž píše dívka Sulla.)

Karel Čapek nacque a Malé Svatonovice nel 1890. Laureatosi in filosofia a Praga, nel 1917–1920 fu redattore di «Fogli nazionali» e del «Giornale del popolo». Cominciò a scrivere racconti insieme al fratello Josef. Karel Čapek morì a Praga nel 1938. Di Čapek va ricordata la vasta produzione drammatica: «RUR» (Rossum’s Universal Robots, 1920) importante nella fantascienza perché per la prima volta è usato il termine di «robot» riferito alla macchina antropomorfa entrato poi nell’uso universale.

Il brano riportato sopra è l’inizio del dramma «RUR», scritto sotto forma di testo teatrale. Il testo è in parentesi perché il personaggio Domin sta parlando al telefono.

As you can note by looking at Table 4, the passage in Czech is put into the environment otherlanguage*. In typing it I have used some abbreviations for inputting characters with diacritics. Czech uses four kinds of diacritics: acute accent, caron, ring and apostrophe. The only letters which can get two of them above them are ‘e’ and ‘u’. The caron over ‘D’ and ‘T’ becomes an apostrophe for the lowercase letter. The acute accent over a vowel denotes the long sound.

Table 4 An example of input in a language different from the main one of the text.

```
\begin{otherlanguage*}{czech}
\textbf{Domin}: (sedí u velikého amerického psacího stolu
v~otá"cecím k"resle. Na stole "zárovka, telefon, t"e"zítka,
po"rada" c dopis"u, atd., na st"en"e vlevo veliké mapy s~lodními
a "zelezni"cními liniemi, velik"y kalendá"r, hodiny, je"z
ukazují n"eco málo p"red polednem; na st"en"e vpravo ti"st"ené
plakáty: "<Nejlacin"ej"ší práce: Rossumovi Roboti"> "<Tropi"ctí
Roboti, nov"y vynález. Kus 150 d."> "<Ka"zd"y si kup svého
Robota!"> "<Chcete zlevnit svoje v"yrobky? Objednejte si
Rossumovy Roboty."> Dále jiné mapy, dopravní lodní "řád,
tabulka s~telegrafick"ymi záznamy kurs"u atd. V~kontrastu
k~tétu v"yzdob"e st"en je na zemi nádhern"y tureck"y koberec,
vpravo kulat"y st"ul, pohovka, ko"žená klubovní k"resla a
knihovna, v~ní"z místo knih stojí láhve s~vínem a ko"ralkami.
Vlevo pokladna. Vedle Dominova stolu psací stroj, na n"em"z
pí"se dívka Sulla.)
\end{otherlanguage*}
```

Here is a table; in the left columns of each block I have written the ‘official’ way of typing the character and in parentheses the ‘unofficial’ one I used (see later).

$\backslash' \{A\}$ (Á) → Á	$\backslash v \{C\}$ ("C) → Č	$\backslash q \{D\}$ ("D) → Ď	$\backslash' \{E\}$ (É) → É
$\backslash v \{E\}$ ("E) → Ě	$\backslash' \{I\}$ (Í) → Í	$\backslash q \{L\}$ ("L) → Ĺ	$\backslash v \{N\}$ ("N) → Ň
$\backslash' \{O\}$ (Ó) → Ó	$\backslash v \{R\}$ ("R) → Ř	$\backslash v \{S\}$ ("S) → Š	$\backslash q \{T\}$ ("T) → Ť
$\backslash' \{U\}$ (Ú) → Ú	$\backslash w \{U\}$ ("U) → Ů	$\backslash' \{Y\}$ ("Y) → Ý	$\backslash v \{Z\}$ ("Z) → Ž
$\backslash' \{a\}$ (á) → á	$\backslash v \{c\}$ ("c) → č	$\backslash q \{d\}$ ("d) → ď	$\backslash' \{e\}$ (é) → é
$\backslash v \{e\}$ ("e) → ě	$\backslash' \{i\}$ (í) → í	$\backslash q \{l\}$ ("l) → ĺ	$\backslash v \{n\}$ ("n) → ň
$\backslash' \{o\}$ (ó) → ó	$\backslash v \{r\}$ ("r) → ř	$\backslash v \{s\}$ ("s) → š	$\backslash v \{t\}$ ("t) → ť
$\backslash' \{u\}$ (ú) → ú	$\backslash w \{u\}$ ("u) → ů	$\backslash' \{y\}$ ("y) → ý	$\backslash v \{z\}$ ("z) → ž

The convention of prefixing a character with a double quote was already used before babel by a package called german on which babel is based. Germans had a very tough time typing their documents in L^AT_EX, and shorthands like "a instead of \{a} were very useful.

Now, let’s come back to the business.

We want the document to be in Italian, but with Czech interspersed in it. Therefore the loading of babel will be through

```
\usepackage[czech,italian]{babel}
```

Remember! The last called language is the main language of the document, so in this case the table of contents would be named *Indice*, as is customary in Italian.

It is not advisable to pass the language option to `\documentclass`, in this case, because of possible bad interaction with the packages. Therefore, if you want to load also `varioref`, for example, you would write

```
\documentclass[a4paper]{report}
...
\usepackage[czech,italian]{babel}
\usepackage[italian]{varioref}
...
```

Sometimes there are strange interactions between packages; a rule of thumb in case incomprehensible error messages are issued by $\text{T}_{\text{E}}\text{X}$ is:

- call every package taking a language option *after* `babel`; this must be done always;
- move *before* `babel` the other packages until the error message is not issued any more;
- call `hyperref` last, unless some package requires explicitly to be called after it.

3 Hyphenation

With $\text{T}_{\text{E}}\text{X}3$ we can hyphenate correctly as many languages as we want, provided the hyphenation rules are made known to it in advance. Well, there is a limit imposed by memory, which can be increased in case of need, and a limit of 256 different set of rules. While this number is less than the number of languages in the world, it should be sufficient for most of us.

The choice of hyphenation rules known to $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ has to be done in advance, when the $\text{T}_{\text{E}}\text{X}$ system is configured. However, all distributions make it possible to rebuild the formats in order to allow for new hyphenation rules. The mechanism specific to a distribution is explained in the distribution's documentation. For $\text{MacT}_{\text{E}}\text{X}$, for example, there is a "Configure" button and one of the things to do is the choice of hyphenation rules. The same applies to $\text{MiK}_{\text{T}}\text{E}_{\text{X}}$ or $\text{T}_{\text{E}}\text{XLive}$.

Some `babel` languages share those rule: `portuges` and `brazil`, for example, or `german` and `austrian`. They are named differently by `babel` because they have different tags in some cases.

Portugueses call *Índice* the Index, while *Brazileiros* call it *Índice Remissivo*. The month of January is *Januar* in *Deutschland*, while it is *Jänner* in *Österreich*.

A very different case is English: hyphenation rules are *not* the same for British English and American English, being them two very different languages.¹ On this account, the rules correspond to distinct choices during the setup.

If hyphenation for a language is not enabled, the rules for English are applied. This can be inconvenient, but there is no provision, up to now, to set things in a different way. In case you are not in charge of the system and want to use, say, Indonesian, a poor person's remedy until the system manager accepts to include the rules for it in the format is to write the following incantation before loading `babel`:

```
\makeatletter
\chardef\l@bahasa=255
\makeatother
```

How to find the exact string to put between `@` and `=` for other languages? In the log file. If you request a language with unknown hyphenation rules, \LaTeX warns you on the screen in this way:

```
No hyphenation patterns were loaded for
the language 'Bahasa Indonesia'
```

You find that message also in the log file and immediately after it you will see the right string to use:

```
\l@bahasa = a dialect from \language0
```

After the incantation, Indonesian will not be hyphenated at all and you will need to cope with bad paragraphs by inserting "soft hyphens", i.e., the command `\-`. This is surely better than getting wrong hyphens. Hopefully, the system manager will be nice enough to enable the language for you: give your thanks, cut off the incantation, erase the soft hyphens and enjoy automatic hyphenation for Indonesian.²

Don't ask what the incantation does; after all, magicians never explain their tricks.

1. G. B. Shaw said that the USA and the UK are two countries divided by a common language. Oscar Wilde said with his characteristic wit "we have really everything in common with America nowadays, except, of course, the language".

2. If *you* are the system manager, then ask some \TeX guru around the discussion forums or, as a last resort, read the manual of your \TeX distribution.

4 Using babel

We have already seen that the last option to the package is considered the main language of the document. For example, this changes automatically the format for dates, when you say `\today`:

US English	March 14, 2007
Czech	14. března 2007
Greek	14 Μαρτίου 2007
Italian	14 marzo 2007
Samin	njukčamánu 14. b. 2007
Ukrainian	14 березня 2007 р.
Welsh	14 Mawrth 2007

Of course you need to input correctly another date, following the standard of the language. Or you can make L^AT_EX work:

```
\newcommand{\printdate}[3]{\%  
  \day=#1\relax\month=#2\relax\year=#3\relax\today}}
```

US English	December 25, 2004
Czech	25. prosince 2004
Greek	25 Δεκεμβρίου 2004
Italian	25 dicembre 2004
Samin	juovlamánu 25. b. 2004
Ukrainian	25 грудня 2004 р.
Welsh	25 Rhagfyr 2004

All these dates are printed giving the same command `\printdate{25}{12}{2004}` in a context where the appropriate language is enforced. If you prefer to give dates in ISO format, there is another incantation:

```
\def\getisodate#1/#2/#3/{%  
  \day=#3\relax\month=#2\relax\year=#1\relax\today}  
\def\printisodate#1{\def\theisodate{#1}%  
  \expandafter\getisodate\theisodate/}}
```

After this we can use something like `\printisodate{1907/5/26}` to get, according to the language, the following results:

US English	May 26, 1907
Czech	26. května 1907
Greek	26 Μαΐου 1907
Italian	26 maggio 1907
Samin	miessemánu 26. b. 1907
Ukrainian	26 травня 1907 р.
Welsh	26 Mai 1907

The package `isodate` has some features to do this thing, if you fear using the magic above. Who was born in that day?

Now we come to the main part of the section: how to change language in the middle of the document.

4.1 A long document with the first part in one language and the second one in another

Suppose you have to write a booklet consisting of four chapters; the first two are in Czech, while the front matter (Introduction), the last two chapters the back matter (an appendix, say) are in Italian. You want that the word ‘Chapter’ comes out in Czech for the first two chapters and in Italian in the last two. In Table 5 there is how to set the document.

If you typeset the sample (changing the dots into some meaningful text), you will notice that the first chapter has the title prefixed by “Kapitola 1”, while the title of the third one will be prefixed by “Capitolo 3”.

To be a bit more formal, the command

```
\selectlanguage{language}
```

changes the complete setup of the document to the chosen *language*, which of course must have been declared at the beginning. All tags after the command `\selectlanguage{czech}` will come out as defined for the Czech language. If the language has also particular typesetting rules, like French, these will be enforced.

In Table 6 you find an example, the same text in French and Italian, with the relative code.

As you can see, in the French text the question mark is preceded by an interword space, as it is customary in French typography, but it is not necessary (although possible) to leave a space in front of it in the input. The same happens for the colon, the semicolon, the exclamation mark and the (French) quotes, but `babel` takes care of these details, not the user. Well, something is left to the user: the ellipsis is typed `...` in the French text, not with `\ldots`.

Table 5 An example of a two language document

```
\documentclass[a4paper]{book}
\usepackage[czech,italian]{babel}
...
\begin{document}
...
\frontmatter
\chapter{Introduzione}
...
\mainmatter
\selectlanguage{czech}
\chapter{Karel \v{C}apek}
...
\chapter{\v{C}eská Republika}
...
\selectlanguage{italian}
\chapter{La vita di Karel \v{C}apek}
...
\chapter{Le opere di Karel \v{C}apek}
...
\appendix
\chapter{Appendice}
...
\end{document}
```

Some languages, notably Hebrew and Arabic, write right to left. Arabic is not supported by babel, but Hebrew is. For switching to right to left typesetting or conversely, the environment `otherlanguage` must be used. Otherwise its effect is the same as `\selectlanguage`. I'm not going to discuss Hebrew typesetting here, I'll only give a code example.

```
\usepackage[hebrew,spanish]{babel}
...
\begin{document}
<text in Spanish>
\begin{otherlanguage}{hebrew}
<text in Hebrew>
\end{otherlanguage}
```

Table 6 The same text in two languages

Qu'est-ce qui a pu faire disjoncter Zizou, dimanche 9 juillet, en finale du Mondial? Qu'est-ce qu'a bien pu lui dire l'Italien Marco Materazzi pour recevoir un coup de tête en plein torse? Depuis la sortie sur carton rouge du capitaine de l'équipe de France, les médias du monde entier se posent la question, tentent de décrypter sur les lèvres de l'Italien la teneur des – supposées – insultes... et les hypothèses se multiplient.

```
\selectlanguage{french}
Qu'est-ce qui a pu faire disjoncter
Zizou, dimanche 9~juillet, en finale
du Mondial? Qu'est-ce qu'a bien pu
lui dire l'Italien Marco Materazzi
pour recevoir un coup de tête en
plein torse? Depuis la sortie sur
carton rouge du capitaine de
l'équipe de France, les médias du
monde entier se posent la question,
tentent de décrypter sur les lèvres
de l'Italien la teneur des
--~supposées~-- insultes... et les
hypothèses se multiplient.
```

Che cosa ha fatto scatenare Zizou, domenica 9 luglio, alla finale dei Mondiali? Che cosa ha potuto dirgli l'italiano Marco Materazzi per ricevere una testata in pieno petto? Dopo l'uscita per il cartellino rosso del capitano della squadra francese, i mezzi di comunicazione del mondo intero si pongono queste domande, tentano di interpretare dalle labbra dell'italiano il tenore dei presunti insulti... e le ipotesi si moltiplicano.

```
\selectlanguage{italian}
Che cosa ha fatto scatenare Zizou,
domenica 9~luglio, alla finale dei
Mondiali? Che cosa ha potuto dirgli
l'italiano Marco Materazzi per
ricevere una testata in pieno
petto? Dopo l'uscita per il
cartellino rosso del capitano della
squadra francese, i mezzi di
comunicazione del mondo intero si
pongono queste domande, tentano di
interpretare dalle labbra
dell'italiano il tenore dei presunti
insulti\ldots\ e le ipotesi si
moltiplicano.
```

```
<text in Spanish>
\end{document}
```

4.2 Some text only in a different language

Most of the time, however, it is only necessary to type some passages in a language different from the main one. Changing the tags would be useless in this case. For this purpose there are a command and an environment which are perfectly equivalent to one another. They are

```
\begin{otherlanguage*}{<language>}
<text>
\end{otherlanguage*}
```

and

```
\foreignlanguage{<language>}{<text>}
```

The environment form is suitable for longer passages consisting of one or more paragraphs; the command form is best employed for short phrases, in line citations, for example.

The text inside the environment or the argument of the command form can be typed with the shorthands provided by babel.

Let's see an example. We know that in German we can input letters with the *Umlaut* by prefixing them with a double quote; a feature very handy for people using a Latin2 keyboard which misses those symbols.

These are the first words of the *Hymn to Joy* by Schiller, set to music by Beethoven: *Freude, Schöner Götterfunken, Tochter aus Elysium*. They were typed as

```
\foreignlanguage{german}{Freude, Sch"oner
G"otterfunken, Tochter aus Elysium}
```

Of course it is not possible to list here all the shorthands for each language. The user guide to babel lists them all.

One feature particularly interesting is the command `\iflanguage`. Suppose that some language prefers boldface type for emphasis rather than italics. You are writing a document in two languages and don't want to use different commands for the same logical structure. So, assume that in Italian we want boldface for emphasis and in English, the main language, we want italics: we define a command in this way

```
\newcommand{\Emph}{\iflanguage{italian}{\textbf}{\emph}}
```

After this we can write something like what is in Table 7. No, in Italian we don't use italics for emphasis, should we?

Table 7 Example of `\iflanguage`

It's easy to put <i>emphasis</i> on words.	È facile mettere parole in evidenza .
<code>\selectlanguage{english}</code> It's easy to put <code>\Emph{emphasis}</code> on words.	<code>\selectlanguage{italian}</code> È facile mettere parole in <code>\Emph{evidenza}</code> .

More programming is possible by taking into account that the current language name is available through the command `\language`. Some more specialized commands are available, but they are not for the common user.

5 Adding or changing tags

Sometimes the tags provided by default just don't fit a user's needs. Suppose that instead of "Figura" as the tag for the figures you want "Illustrazione", for an article with Italian as its main language. The correct way to cope with this is *not* to redefine the command `\figurename` that, you have guessed, contains the tag.

The problem is that, after a language shift with `\selectlanguage` and a return to Italian, the figures will be labeled again with "Figura". This happens because `babel` keeps a list of those tags and does a bunch of `\renewcommand` stuff, but the original tag remains.

It's not difficult to arrange this; the tags for Italian are restored with the command `\captionsitalian` and so all we need is to add our definition to it:

```
\addto\captionsitalian{\renewcommand{\figurename}{Illustrazione}}
```

in the preamble will set everything as we want.

It is in order to give a list of all predefined tags. You find it in Table 8. Their meaning can be changed in the same way as in the example before, on a language basis.

You can also add tags of your own. Say that you need a tag for algorithms. This should be "Algorithm" in English and "Algoritmo" in Italian. The method is analogous to the one for changing, but first we have to make the symbolic name known to `LATEX`; therefore we write in the preamble:

Table 8 The list of predefined tags, with their value in English

<code>\prefacename</code> (Preface)	<code>\tablename</code> (Table)
<code>\refname</code> (References)	<code>\partname</code> (Part)
<code>\abstractname</code> (Abstract)	<code>\enclname</code> (encl)
<code>\bibname</code> (Bibliography)	<code>\ccname</code> (cc)
<code>\chaptername</code> (Chapter)	<code>\headtoname</code> (To)
<code>\appendixname</code> (Appendix)	<code>\pagename</code> (Page)
<code>\contentsname</code> (Contents)	<code>\seename</code> (see)
<code>\listfigurename</code> (List of Figures)	<code>\alsoname</code> (see also)
<code>\listtablename</code> (List of Tables)	<code>\proofname</code> (Proof)
<code>\indexname</code> (Index)	<code>\glossaryname</code> (Glossary)
<code>\figurename</code> (Figure)	

```
\newcommand{\algorithmname}{}  
\addto\captionenglish{\renewcommand{\algorithmname}{Algorithm}}  
\addto\captionitalian{\renewcommand{\algorithmname}{Algoritmo}}
```

Of course you will add the tag definition to every language for which you need to use that new tag.