Babel, a multilingual package for use with LaTeX's standard document classes*

Johannes Braams
Kooienswater 62
2715 AJ Zoetermeer
The Netherlands
J.L.Braams@cistron.nl

Printed July 26, 1995

Abstract

The standard distribution of LaTeX contains a number of document classes that are meant to be used, but also serve as examples for other users to create their own document classes. These document classes have become very popular among LaTeX users. But it should be kept in mind that they were designed for American tastes and typography. At one time they contained a number of hard-wired texts. This report describes babel, a package that makes use of the new capabilities of TeX version 3 to provide an environment in which multilingual documents can be written.

Contents

1 The user interface **2**

2 Changes for LaTeX 2ε **3**

3 Changes in Babel version 3.5 **3**

4 The interface between the core of babel and the language definition files **4**

4.1 Support for active characters 5

4.2 Support for saving macro definitions 5

4.3 Support for extending macros 6

4.4 Macros common to a number of languages . . . 6

5 Compatibility with german.sty **6**

6 Conclusion **8**

7 Acknowledgements **8**

8 The Esperanto language **9**

9 The Dutch language **9**

10 The English language **9**

11 The German language **10**

12 The Breton language **11**

13 The Irish language **11**

14 The scottish language **11**

15 The French language **11**

*During the development ideas from Nico Poppelier, Piet van Oostrum and many others have been used. Bernd Raichle has provided many helpful suggestions.

16 The Italian language 12
17 The Portuguese language 12
18 The Spanish language 12
19 The Catalan language 14
20 The Galician language 15
21 The Romanian language 15
22 The Danish language 16
23 The Norwegian language 16
24 The Swedish language 16
25 The Finnish language 16
26 The Hungarian language 17
27 The Estonian language 17
28 The Croatian language 18
29 The Czech language 18
30 The Polish language 19
31 The Slovak language 19
32 The Slovanian language 20
33 The Lower Sorbian language 20
34 The Upper Sorbian language 20
35 The Turkish language 20
36 The Bahasa language 21

1 The user interface

The user interface of this package is quite simple. It basially consists of only two commands. These commands can be used to select another language or to find out what the current language is.

`language` The environment `language` does basically the same as `\selectlanguage`, except the language change is local to the environment. For mixing left-to-right typesetting with right-to-left typesetting the use of this environment is a prerequisite.

`\foreignlanguage` The command `\foreignlanguage` takes two arguments, the second argument is a phrase to be typeset according to the rules of the language named in its first argument.

`\selectlanguage` When a user wants to switch from one language to another he can do so using the macro `\selectlanguage`. This macro takes the language, defined previously by a language definition file, as its argument. It calls several macros that should be defined in the language definition files to activate the special definitions for the language chosen.

`\languagename` The name of the current language is stored in the control sequence `\languagename`.

`\iflanguage` If more than one language is used it might be necessary to know which language is active at a specific time. This can be checked by a call to `\iflanguage`. This macro takes three arguments. The first argument is the name of a language, the second and third arguments are the actions to take if the result of the test is `true` or `false` respectively.

`\useshorthands` A command with one argument, the character to make active in order to define personal shorthands.

`\defineshorthand` The command `\defineshorthand` takes two arguments, the first of which is a one or two character sequence, the second argument is the code the shorthand should expand to.

\languageshorthands

The command \languageshorthands can be used to switch the shorthands on the language level. It takes one argument, the name of a language. Note that for this to work the language should have been specified as an option when loading the babel package.

2 Changes for LaTeX 2ε

With the advent of LaTeX 2ε the interface to babel in the preamble of the doument has changed. With LaTeX2.09 one used to call up the babel system with a line such as:

```
\documentstyle[dutch,english]{article}
```

which would tell LaTeX that the document would be written in two languages, dutch and english and that english would be the first language in use.

The LaTeX2e way of providing the same information is:

```
\documentclass{article}
\usepackage[dutch,english]{babel}
```

or, making dutch and english global options in order to let other packages detect and use them:

```
\documentclass[dutch,english]{article}
\usepackage{babel}
\usepackage{varioref}
```

In this last example the package varioref will also see the options and will be able to use them.

3 Changes in Babel version 3.5

In Babel version 3.5 a lot of changes have been made when compared with the previous release. Here is a list of the most important ones:

- babel now has a language environment and a new command \foreignlanguage;
- the way active characters are dealt with is completely changed. They are called 'shorthands'; one can have three levels of shorthands: on the user level, the language level and on 'system level'. A consequence of the new way of handling active characters is that they are now written to auxiliary files 'verbatim';
- A language change now also writes information in the .aux file as the change might also affect typesetting the table of contents. The consequence is that an .aux file generated by a LaTeX format with babel preloaded gives errors when read with a LaTeX format without babel, but I think this problaly doesn't occur;
- babel is now compatible with the inputenc and fontenc packages;
- the language definition files now have a new extension, ldf;

- the syntax of the file `language.dat` is extended to be compatible with the `french` package by Bernard Gaulle;
- each language definition file looks for a configuration file which has the same name, but the extension `.cfg`.

4 The interface between the core of babel and the language definition files

In the core of the babel system two macros are defined that are to be used in language definition files. Their purpose is to make a new language known.

`\addlanguage` The macro `\addlanguage` is a non-outer version of the macro `\newlanguage`, defined in `plain.tex` version 3.x. For older versions of `plain.tex` and `lplain.tex` a substitute definition is used.

`\adddialect` The macro `\adddialect` can be used in the case where two languages can (or have to) use the same hyphenation patterns. This can be useful when a user wants to use a language for which no patterns are preloaded in the format. In such a case the default behaviour of the babel system is to define this language as a 'dialect' of the language for which the patterns were loaded as `\language0`.

The language definition files have to conform to a number of conventions. The reason for this is that these files have to fill in the gaps left by the common code in `babel.def`, i. e., the definitions of the macros that produce texts. Also the language-switching possibility which has been built into the babel system has its implications.

The following assumptions are made:

- Some of the language-specific definitions might be used by plain TEX users, so the files have to be coded such that they can be read by LATEX as well as by plain TEX. This can be checked by looking at the value of the macro `\fmtname`.
- The common part of the babel system redefines a number of macros and environments (defined previously in the document style) to put in the names of macros that replace the previously hard-wired texts. These macros have to be defined in the language definition files.
- The language definition files define five macros, used to activate and deactivate the language-specific definitions. These macros are `\`⟨*lang*⟩`hyphenmins`, `\captions`⟨*lang*⟩, `\date`⟨*lang*⟩, `\extras`⟨*lang*⟩and `\noextras`⟨*lang*⟩. These macros and their functions are discussed below.
- When a language definition file is loaded, it can define `\l@`⟨*lang*⟩to be a dialect of `\language0` when `\l@`⟨*lang*⟩is undefined.
- The languagedefinition files can be read in the preamble of the document, but also in the middle of document processing. This means that they have to function independently of the current `\catcode` of the `@` sign.

`\langhyphenmin` The macro `\`⟨*lang*⟩`hyphenmins` is used to store the values of the `\lefthyphenmin` and `\righthyphenmin`.

`\captionslang` The macro `\captions`⟨*lang*⟩defines the macros that hold the texts to replace

the original hard-wired texts.

\datelang The macro \date⟨*lang*⟩defines \today and

\extraslang The macro \extras⟨*lang*⟩contains all the extra definitions needed for a specific language.

\noextraslang Because we want to offer the user the possibility to switch between languages and we do not know in what state TEX might be after the execution of \extras⟨*lang*⟩, a macro that brings TEX into a predefined state is needed. It will be no surprise that the name of this macro is \noextras⟨*lang*⟩.

\main@language To postpone the activation of the definitions needed for a language untill the beginning of a document, all language definition files should use \main@language instead of \selectlanguage. This will just store the name of the language and the proper language will be activated at the start of the document.

\loadlocalcfg At the end of the processing of a language definition file LATEX can be instructed to load a local configuration file. This file can for instance be used to add strings to \captions⟨*lang*⟩ in order to support local document classes. The user will be informed of the fact that this configuration file is loaded.

4.1 Support for active characters

In quite a few language definition files, active characters are introduced. To facilitate this, some support macros are provided.

\initiate@active@char The internal macro \initiate@active@char is used in language definition files to instruct LATEX to give a character the category code 'active'. When a character has been made active it will remain that way untill the end of then document. Its definition may vary.

\bbl@activate The command \bbl@activate is used to change the way an active character
\bbl@deactivate expands. \bbl@activate 'switches on' the active behaviour of the character. \bbl@deactive lets the active character expand to its former (mostly) non-active self.

\declare@shorthand The macro \declare@shorthand is used to define the various shorthands. It takes three arguments, the name for the collection of shorthands this definition belongs to; the character (sequence) that makes up the shorthand i.i. ~ or "a and the code to be executed when the shorthand is encountered.

\bbl@add@special "Plain TEX includes a macro called \dospecials that is essentially a set macro,
\bbl@remove@special representing the set of all characters that have a special category code." [1, p. 380] It is used to set text 'verbatim'. To make this work if more characters get a special category code, you have to add this character to the macro \dospecial. LATEX adds another macro called \@sanitize representing the same character set, but without the curly braces. The macros \bbl@add@special⟨*char*⟩ and \bbl@remove@special⟨*char*⟩ add and remove the character ⟨*char*⟩ to these two sets.

4.2 Support for saving macro definitions

Language definition files may want to *re*define macros that already exist. Therefore a mechanism for saving (and restoring) the original definition of those macros is provided. We provide two macros for this[1].

\babel@save To save the current meaning of any control sequence the macro \babel@save

[1] This mechanism was introduced by Bernd Raichle.

is provided. It takes one argument, ⟨*csname*⟩, the control sequence for which the meaning has to be saved.

`\babel@savevariable` A second macro is provided to save the current value of a variable. In this context anything that is allowed after the `\the` primitive is considered to be a variable. The macro takes one argument, the ⟨*variable*⟩.

The effect of the aforementioned macros is that a piece of code is appended to the current definition of `\originalTeX`. When `\originalTeX` is expanded this code restores the previous definition of the control sequence or the previous value of the variable.

4.3 Support for extending macros

`\addto` The macro `\addto{`⟨*control sequence*⟩`}{`⟨*TEX code*⟩`}` can be used to extend the definition of a macro. The macro need not be defined. This macro can, for instance, be used in adding instructions to a macro like `\extrasenglish`.

4.4 Macros common to a number of languages

`\allowhyphens` In a couple of european languages compound words are used. This means that when TEX has to hyphenate such a compound word it only does that at the '-' that is used in such words. To allow hyphenation in the rest of such a compound word the macro `\allowhyphens` can be used.

`\set@low@box` For some languages quotes need to be lowered to the baseline. For this purpose the macro `\set@low@box` is available. It takes one argument and puts that argument in an `\hbox`, at the baseline. The result is available in `\box0` for further processing.

`\save@sf@q` Sometimes it is necessary to preserve the `\spacefactor`. For this purpose the macro `\save@sf@q` is available. It takes one argument, saves the current spacefactor, executes the argument and restores the spacefactor.

`\bbl@frenchspacing` `\bbl@nonfrenchspacing` The commands `\bbl@frenchspacing` and `\bbl@nonfrenchspacing` can be used to properly switch french spacing on and off.

5 Compatibility with `german.sty`

As has been discussed before, the file `german.sty` has been one of the sources of inspiration for the babel system. Because of this I wanted to include `german.sty` in the babel system. To be able to do that I had to allow for one incompatibility: in the definition of the macro `\selectlanguage` in `german.sty` the argument is used as the ⟨*number*⟩ for an `\ifcase`. So in this case a call to `\selectlanguage` might look like `\selectlanguage{\german}`.

In the definition of the macro `\selectlanguage` in `babel.def` the argument is used as a part of other macronames, so a call to `\selectlanguage` now looks like `\selectlanguage{german}`. Notice the absence of the escape character. As of version 3.1a of babel both syntaxes are allowed.

All other features of the original `german.sty` have been copied into a new file, called `germanb.sty`[2].

Although the babel system was developed to be used with LATEX, some of the features implemented in the language definition files might be needed by plain

[2] The 'b' is added to the name to distinguish the file from Partls' file.

TEX users. Care has been taken that all files in the system can be processed by plain TEX.

6 Conclusion

A system of document options has been presented that enable the user of LaTeX to adapt the standard document classes of LaTeX to the language he or she prefers to use. These options offer the possibility to switch between languages in one document. The basic interface consists of using ones option, which is the same for *all* standard document classes.

In some cases the language definition files provide macros that can be of use to plain TeX users as well as to LaTeX users. The babel system has been implemented in such a way that it can be used by both groups of users.

7 Acknowledgements

I would like to thank all who volunteered as β-testers for their time. I would like to mention Julio Sanchez who supplied the option file for the Spanish language and Maurizio Codogno who supplied the option file for the Italian language. Werenfried Spit supplied the files for the Russian language. Michel Goossens supplied contributions for most of the other languages. Nico Poppelier helped polishing the text of the documentation and supplied parts of the macros for the Dutch language. Paul Wackers and Werenfried Spit helped finding and repairing bugs.

During the further development of the babel system I received much help from Bernd Raichle, for which I am grateful.

References

[1] Donald E. Knuth, *The TeXbook*, Addison-Wesley, 1986.

[2] Leslie Lamport, *LaTeX, A document preparation System*, Addison-Wesley, 1986.

[3] K.F. Treebus. *Tekstwijzer, een gids voor het grafisch verwerken van tekst.* SDU Uitgeverij ('s-Gravenhage, 1988). A Dutch book on layout design and typography.

[4] Hubert Partl, *German TeX*, *TUGboat* 9 (1988) #1, p. 70–72.

[5] Leslie Lamport, in: TeXhax Digest, Volume 89, #13, 17 februari 1989.

[6] Johannes Braams, Victor Eijkhout and Nico Poppelier, *The development of national LaTeX styles*, *TUGboat* 10 (1989) #3, p. 401–406.

[7] Joachim Schrod, *International LaTeX is ready to use*, *TUGboat* 11 (1990) #1, p. 87–90.

8 The Esperanto language

The file `esperant.dtx`[3] defines all the language-specific macros for the Esperanto language.

For this language the character `^` is made active. In table 1 an overview is given of its purpose.

`^u`	gives ŭ, with hyphenation in the rest of the word allowed
`^U`	gives Ŭ, with hyphenation in the rest of the word allowed
`^h`	prevents ĥ from becoming too tall
`^j`	gives ĵ
`^|`	inserts a `\discretionary{-}{}{}`
`^a`	gives â with hyphenation in the rest of the word allowed, this works for all other letters as well.

Table 1: The funtions of the active character for Esperanto.

9 The Dutch language

The file `dutch.dtx`[4] defines all the language-specific macros for the Dutch language.

For this language the character `"` is made active. In table 2 an overview is given of its purpose. One of the reasons for this is that in the Dutch language a word with a dieresis can be hyphenated just before the letter with the umlaut, but the dieresis has to disappear if the word is broken between the previous letter and the accented letter.

In [3] the quoting conventions for the Dutch language are discussed. The preferred convention is the single-quote Anglo-American convention, i.e. 'This is a quote'. An alternative is the slightly old-fashioned Dutch method with initial double quotes lowered to the baseline, „This is a quote", which should be typed as `"'This is a quote"'`.

10 The English language

The file `english.dtx`[5] defines all the language definition macros for the English language as well as for the American version of this language.

For this language currently no special definitions are needed or available.

[3] The file described in this section has version number ? and was last revised on ?. A contribution was made by Ruiz-Altaba Marti (`ruizaltb@cernvm.cern.ch`). Code from the file `esperant.sty` by Jörg Knappen (`knappen@vkpmzd.kph.uni-mainz.de`) was included.

[4] The file described in this section has version number ?, and was last revised on ?.

[5] The file described in this section has version number ? and was last revised on ?.

`"a`	`\"a` which hyphenates as `-a`; also implemented for the other letters.
`"y`	puts a negative kern between `i` and `j`
`"Y`	puts a negative kern between `I` and `J`
`"|`	disable ligature at this position.
`"-`	an explicit hyphen sign, allowing hyphenation in the rest of the word.
`"‘`	lowered double left quotes (see example below).
`"’`	normal double right quotes.
`\-`	like the old `\-`, but allowing hyphenation in the rest of the word.

Table 2: The extra definitions made by `dutch.ldf`

11 The German language

The file `germanb.dtx`[6] defines all the language definition macros for the German language as well as for the Austrian dialect of this language[7].

For this language the character " is made active. In table 3 an overview is given of its purpose. One of the reasons for this is that in the German language some character combinations change when a word is broken between the combination. Also the vertical placement of the umlaut can be controlled this way. The quotes

`"a`	`\"a`, also implemented for the other lowercase and uppercase vowels.
`"s`	to produce the German ß (like `\ss{}`).
`"z`	to produce the German ß (like `\ss{}`).
`"ck`	for `ck` to be hyphenated as `k-k`.
`"ff`	for `ff` to be hyphenated as `ff-f`, this is also implemented for l, m, n, p, r and t
`"S`	for `SS` to be `\uppercase{"s}`.
`"Z`	for `SZ` to be `\uppercase{"z}`.
`"|`	disable ligature at this position.
`"-`	an explicit hyphen sign, allowing hyphenation in the rest of the word.
`""`	like `"-`, but producing no hyphen sign (for compund words with hyphen, e.g. `x-""y`).
`"~`	for a compound word mark without a breakpoint.
`"=`	for a compound word mark with a breakpoint, allowing hyphenation in the composing words.
`"‘`	for German left double quotes (looks like ,,).
`"’`	for German right double quotes.
`"<`	for French left double quotes (similar to <<).
`">`	for French right double quotes (similar to >>).

Table 3: The extra definitions made by `german.ldf`

[6] The file described in this section has version number ? and was last revised on ?.

[7] This file is a re-implementation of Hubert Partl's `german.sty` version 2.5b, see [4].

in table 3 can also be typeset by using the commands in table 4.

`\glqq`	for German left double quotes (looks like ,,).
`\grqq`	for German right double quotes (looks like “).
`\glq`	for German left single quotes (looks like ,).
`\grq`	for German right single quotes (looks like ‘).
`\flqq`	for French left double quotes (similar to <<).
`\frqq`	for French right double quotes (similar to >>).
`\flq`	for (French) left single quotes (similar to <).
`\frq`	for (French) right single quotes (similar to >).
`\dq`	the original quotes character (").

Table 4: More commands which produce quotes, defined by `german.ldf`

12 The Breton language

The file `breton.dtx`[8] defines all the language-specific macros for the Breton language.

There are not really typographic rules for the Breton language. It is a local language (it's one of the celtic languages) which is spoken in Brittany (West of France). So we have a synthesis between french typographic rules and english typographic rules. The characters `:`, `;`, `!` and `?` are made active in order to get a whitespace automatically before these characters.

13 The Irish language

The file `irish.dtx`[9] defines all the language definition macros for the Irish language.

14 The scottish language

The file `scottish.dtx`[10] defines all the language definition macros for the scottish language.

15 The French language

The file `francais.dtx`[11] defines all the language definition macros for the French language.

[8] The file described in this section has version number ? and was last revised on ?.

[9] The file described in this section has version number ? and was last revised on ?. A contribution was made by Marion Gunn.

[10] The file described in this section has version number ? and was last revised on ?. A contribution was made by Fraser Grant (`FRASER@CERNVM`).

[11] The file described in this section has version number ? and was last revised on ?. This file was initially derived from the original version of `german.sty`, which has some definitions for French. Later the definitions from `french.sty` version 2 were added.

French typographic rules specify that a little white space should be present before 'double puctuation' characters. These characters are `:`, `;`, `!` and `?`. In order to get this whitespace automatigically the category code of these characters is made `\active`. The user should input these four characters preceeded with a space; the space will then be replaced by a `\thinspace`.

16 The Italian language

The file `italian.dtx`[12] It defines all the language-specific macros for the Italian language.

For this language the `\clubpenalty`, `\widowpenalty` and `\finalhyphendemerits` are set to rather high values.

17 The Portuguese language

The file `portuges.dtx`[13] defines all the language-specific macros for the Portuguese language as well as for the Brasilian version of this language.

For this language the character `"` is made active. In table 5 an overview is given of its purpose.

`"|`	disable ligature at this position.
`"-`	an explicit hyphen sign, allowing hyphenation in the rest of the word.
`""`	like `"-`, but producing no hyphen sign (for words that should break at some sign such as "entrada/salida."
`"<`	for French left double quotes (similar to <<).
`">`	for French right double quotes (similar to >>).
`\-`	like the old `\-`, but allowing hyphenation in the rest of the word.

Table 5: The extra definitions made by `portuges.ldf`

18 The Spanish language

The file `spanish.dtx`[14] defines all the language definition macro's for the Spanish[15] language.

[12] The file described in this section has version number ? and was last revised on ?. The original author is Maurizio Codogno, (`urcm@ur785.cselt.stet.it`).

[13] The file described in this section has version number ? and was last revised on ?. Contributions were made by Jose Pedro Ramalhete (`JRAMALHE@CERNVM` or `Jose-Pedro_Ramalhete@MACMAIL`) and Arnaldo Viegas de Lima `arnaldo@VNET.IBM.COM`.

[14] The file described in this section has version number ? and was last revised on ?. The original author is Julio Sánchez, (`jsanchez@gmv.es`).

[15] Catalan used to be part of this file but is now on its own file.

This file[16] incorporates the result of discussions held in the Spanish-TEX[17] electronic mail list.

For this language the characters ' ~ and " are made active. In table 6 an overview is given of their purpose. These active accent characters behave according

`'a`	an accent that allows hyphenation. Valid for all vowels uppercase and lowercase.
`'n`	a n with a tilde. This is included to improve compatibility with FTC. Works for uppercase too.
`"|`	disable ligature at this position.
`"-`	an explicit hyphen sign, allowing hyphenation in the rest of the word.
`""`	like `"-`, but producing no hyphen sign (for words that should break at some sign such as "entrada/salida."
`\-`	like the old `\-`, but allowing hyphenation in the rest of the word.
`"u`	a u with dieresis allowing hyphenation.
`"a`	feminine ordinal as in 1ª.
`"o`	masculine ordinal as in 1º.
`"<`	for French left double quotes (similar to <<).
`">`	for French right double quotes (similar to >>).
`~n`	a n with tilde. Works for uppercase too.

Table 6: The extra definitions made by `spanish.ldf`

to their original definitions if not followed by one of the characters indicated in that table.

This option includes support for working with extended, 8-bit fonts, if available. Old versions of this file based this support on the existance of special macros with names as in Ferguson's ML-TEX. This is no longer the case. Support is now based on providing an appropriate definition for the accent macros on entry to the Spanish language. This is automatically done by LaTeX 2_ε or NFSS2. If T1 encoding is chosen, and provided that adequate hyphenation patterns[18] exist, it is possible to get better hyphenation for Spanish than before. The easiest way to use the new encoding with LaTeX 2_ε to load the package `t1enc` with `\usepackage`. This must be done before loading babel.

[16] In writing this file, many ideas and actual coding solutions have been taken from a number of sources. The language definition files `dutch.sty` and `germanb.sty` are the main contributors and are not explicitly mentioned in the sequel. J. L. Braams and Bernd Raichle have given helpful advice. Another source of inspiration is the experience gained in the use of FTC, a software package written by José A. Mañas. The members of the Spanish-TEX list have helped clarify a number of issues. Other sources are explicitly acknowledged when used. If you think that you contributed something and you are not mentioned, please let me (`jsanchez@gmv.es`) know. I humbly apologize for any omission.

[17] `spanish-tex@goya.eunet.es`, subscription requests can be sent to the address `listserv@goya.eunet.es`. This list is devoted to discussions on support in TEX for Spanish. Comments on this language option are welcome there or directly to `jsanchez@gmv.es`.

[18] One source for such patterns is the archive at `ftp.eunet.es` that can be accessed by anonymous FTP or electronic mail to `ftpmail@goya.eunet.es`. They are in the `info` directory `src/TeX/spanish`. The list of Frequently Asked Questions with Answers about TEX for Spanish is kept there as well. That list is meant to be a summary of the discussions held in the Spanish-TEX mail list. Warning: It is in Spanish.

If the combination of keyboard and TEX version that the user has is able to produce the accented characters in the T1 enconding, the user could see the accented characters in the editor, greatly improving the readability of the document source. As of today, this is not a recommended method for producing documents for distribution, although it is possible to mechanically translate the document so that the receiver can make use of it. If care is taken to define the encoding needed by the document, the results are pretty portable.

This option file will automatically detect if the T1 encoding is being used and behave appropriately. If any other encoding is being used, the accent macros will be redefined to allow hyphenation on the accented words.

19 The Catalan language

The file `catalan.dtx`[19] defines all the language-specific macro's for the Catalan language.

For this language only the double quote character (") is made active by default. In table 7 an overview is given of the new macros defined and the new meanings of ". Additionally to that, the user can explicitly activate the acute accent or apostrophe (') and/or the grave accent (‘) characters by using the activeacute and activegrave options. In that case, the definitions shown in table 8 become also available[20].

`\lgem`	geminated-l digraph (similar to l·l). `\Lgem` produces the uppercase version.
`\up`	Macro to help typing raised ordinals, like 1^{er}. Takes one argument.
`\-`	like the old `\-`, but allowing hyphenation in the rest of the word.
`"i`	i with diaeresis, allowing hyphenation in the rest of the word. Valid for the following vowels: i, u (both lowercase and uppercase).
`"c`	c-cedilla (ç). Valid for both uppercase and lowercase c.
`"l`	geminated-l digraph (similar to l·l). Valid for both uppercase and lowercase l.
`"<`	French left double quotes (similar to <<).
`">`	French right double quotes (similar to >>).
`"-`	explicit hyphen sign, allowing hyphenation in the rest of the word.
`"|`	disable ligature at this position.

Table 7: Extra definitions made by file `catalan.ldf` (activated by default)

These active accents characters behave according to their original definitions if not followed by one of the characters indicated in that table.

[19] The file described in this section has version number ? and was last revised on ?.

[20] Please note that if the acute accent character is active, it is necessary to take special care of coding apostrophes in a way which cannot be confounded with accents. Therefore, it is necessary, for example, to type `l'{}{}astre` instead of `l'astre`.

`'e`	acute accented a, allowing hyphenation in the rest of the word. Valid for the following vowels: e, i, o, u (both lowercase and uppercase).
`‘a`	grave accented a, allowing hyphenation in the rest of the word. Valid for the following vowels: a, e, o (both lowercase and uppercase).

Table 8: Extra definitions made by file `catalan.ldf` (activated only when using the options activeacute and activegrave)

20 The Galician language

The file `galician.dtx`[21] defines all the language definition macros for the Galician language.

For this language the characters ' ~ and " are made active. In table 9 an overview is given of their purpose. These active accents character behave according

`"|`	disable ligature at this position.
`"-`	an explicit hyphen sign, allowing hyphenation in the rest of the word.
`\-`	like the old \-, but allowing hyphenation in the rest of the word.
`'a`	an accent that allows hyphenation. Valid for all vowels uppercase and lowercase.
`'n`	a n with a tilde. This is included to improve compatibility with FTC. Works for uppercase too.
`"u`	a u with dieresis allowing hyphenation.
`"a`	feminine ordinal as in 1ª.
`"o`	masculine ordinal as in 1º.
`~n`	a n with tilde. Works for uppercase too.

Table 9: The extra definitions made by `galician.ldf`

to their original definitions if not followed by one of the characters indicated in that table.

21 The Romanian language

The file `romanian.dtx`[22] defines all the language-specific macros for the Romanian language.

For this language currently no special definitions are needed or available.

[21] The file described in this section has version number ? and was last revised on ?.

[22] The file described in this section has version number ? and was last revised on ?. A contribution was made by Umstatter Horst (`hhu@cernvm.cern.ch`).

22 The Danish language

The file `danish.dtx`[23] defines all the language definition macros for the Danish language.

For this language the character " is made active. In table 10 an overview is given of its purpose.

`"|`	disable ligature at this position.
`"-`	an explicit hyphen sign, allowing hyphenation in the rest of the word.
`""`	like `"-`, but producing no hyphen sign (for words that should break at some sign such as "entrada/salida."
`"‘`	lowered double left quotes (looks like ,,)
`"’`	normal double right quotes
`"<`	for French left double quotes (similar to <<).
`">`	for French right double quotes (similar to >>).

Table 10: The extra definitions made by `danish.ldf`

23 The Norwegian language

The file `norsk.dtx`[24] defines all the language definition macros for the Norwegian language as well as for a new spelling variant 'nynorsk' for this language.

For this language currently no special definitions are needed or available.

24 The Swedish language

The file `swedish.dtx`[25] defines all the language-specific macros for the Swedish language.

For this language currently no special definitions are needed or available.

25 The Finnish language

The file `finnish.dtx`[26] defines all the language definition macros for the Finnish language.

For this language the character " is made active. In table 11 an overview is given of its purpose.

[23] The file described in this section has version number ? and was last revised on ?. A contribution was made by Henning Larsen (`larsen@cernvm.cern.ch`)

[24] The file described in this section has version number ? and was last revised on ?. Contributions were made by Haavard Helstrup (`HAAVARD@CERNVM`) and Alv Kjetil Holme (`HOLMEA@CERNVM`); the 'nynorsk' variant has been supplied by Per Steinar Iversen `iversen@vxcern.cern.ch`) and Terje Engeset Petterst (`TERJEEP@VSFYS1.FI.UIB.NO`).

[25] The file described in this section has version number ? and was last revised on ?. A contribution was made by Sten Hellman (`HELLMAN@CERNVM.CERN.CH`).

[26] The file described in this section has version number ? and was last revised on ?. A contribution was made by Mikko KANERVA (`KANERVA@CERNVM`) and Keranen Reino (`KERANEN@CERNVM`).

"|	disable ligature at this position.
"-	an explicit hyphen sign, allowing hyphenation in the rest of the word.
"=	an explicit hyphen sign for expressions such as "pakastekaapit ja -arkut".
""	like "-, but producing no hyphen sign (for words that should break at some sign such as "entrada/salida."
"‘	lowered double left quotes (looks like ,,)
"’	normal double right quotes
"<	for French left double quotes (similar to <<).
">	for French right double quotes (similar to >>).
\-	like the old \-, but allowing hyphenation in the rest of the word.

Table 11: The extra definitions made by `finnish.ldf`

26 The Hungarian language

The file option `magyar.dtx`[27] defines all the language definition macros for the Hungarian language.

`\ontoday` For this language currently the only special definition that is added is the `\ontoday` command which works like `\today` but produces a slightly different date format used in expressions suh as 'on february 10th'.

27 The Estonian language

The file `estonian.dtx`[28] defines the language definition macro's for the Estonian language.

This file was written as part of the TWGML project, and borrows heavily from the babel German and Spanish language files `germanb.ldf` and `spanish.ldf`.

Estonian has the same umlauts as German (ä, ö, ü), but in addition to this, we have also õ, and two recent characters š and ž, so we need at least two active characters. We shall use " and ~ to type Estonian accents on ASCII keyboards (in the 7-bit character world). Their use is given in table 12. These active accent characters behave according to their original definitions if not followed by one of the characters indicated in that table; the original quote character can be typed using the macro `\dq`.

We support also the T1 output encoding (and Cork-encoded text input). You can choose the T1 encoding by the command `\usepackage[T1]{fontenc}`. This package must be loaded before babel. As the standard Estonian hyphenation file `eehyph.tex` is in the Cork encoding, choosing this encoding will give you better hyphenation.

As mentioned in the Spanish style file, it may happen that some packages

[27] The file described in this section has version number ? and was last revised on ?. A contribution was made by Attila Koppanyi (`attila@cernvm.cern.ch`). Later updates and suggestions by Árpád Bíró (`JZP1104@HUSZEG11.bitnet`), Istvan Hamecz (`hami@ursus.bke.hu`) and Horvath Dezso (`horvath@pisa.infn.it`).

[28] The file described in this section has version number ? and was last revised on ?. The original author is Enn Saar, (`saar@aai.ee`).

~o	\~o, (and uppercase);
"a	\"a, (and uppercase);
"o	\"o, (and uppercase);
"u	\"u, (and uppercase);
~s	\v s, (and uppercase);
~z	\v z, (and uppercase);
"|	disable ligature at this position;
"-	an explicit hyphen sign, allowing hyphenation in the rest of the word;
\-	like the old \-, but allowing hyphenation in the rest of the word;
"‘	for Estonian low left double quotes (same as German);
"’	for Estonian right double quotes;
"<	for French left double quotes (also rather popular)
">	for French right double quotes.

Table 12: The extra definitions made by `estonian.ldf`

fail (usually in a `\message`). In this case you should change the order of the `\usepackage` declarations or the order of the style options in `\documentclass`.

28 The Croatian language

The file `croatian.dtx`[29] defines all the language definition macros for the Croatian language.

For this language currently no special definitions are needed or available.

29 The Czech language

The file `czech.dtx`[30] defines all the language definition macros for the Czech language.

For this language `\frenchspacing` is set and two macros `\q` and `\w` for easy access to two accents are defined.

The command `\q` is used with the letters (`t`, `d`, `l`, and `L`) and adds a `’` to them to simulate a ‘hook’ that should be there. The result looks like ť. The command `\w` is used to put the ring-accent which appears in ångstrøm over the letters `u` and `U`.

[29] The file described in this section has version number ? and was last revised on ?. A contribution was made by Alan Paić (`paica@cernvm.cern.ch`).

[30] The file described in this section has version number ? and was last revised on ?. Contributions were made by Milos Lokajicek (`LOKAJICK@CERNVM`).

30 The Polish language

The file `polish.dtx`[31] defines all the language-specific macros for the Polish language.

For this language the character " is made active. In table 13 an overview is given of its purpose.

`"a`	or `\aob`, for tailed-a (like ą)
`"A`	or `\Aob`, for tailed-A (like Ą)
`"e`	or `\eob`, for tailed-e (like ę)
`"E`	or `\Eob`, for tailed-E (like Ę)
`"c`	or `\'c`, for accented c (like ć), same with uppercase letters and n,o,s
`"l`	or `\lpb{}`, for l with stroke (like ł)
`"L`	or `\Lpb{}`, for L with stroke (like Ł)
`"r`	or `\zkb{}`, for pointed z (like ż), cf. pronounciation
`"R`	or `\Zkb{}`, for pointed Z (like Ż)
`"z`	or `\'z`, for accented z
`"Z`	or `\'Z`, for accented Z
`"|`	disable ligature at this position.
`"-`	an explicit hyphen sign, allowing hyphenation in the rest of the word.
`""`	like `"-`, but producing no hyphen sign (for compund words with hyphen, e.g. `x-""y`).
`"‘`	for German left double quotes (looks like ,,).
`"’`	for German right double quotes.
`"<`	for French left double quotes (similar to <<).
`">`	for French right double quotes (similar to >>).

Table 13: The extra definitions made by `polish.sty`

31 The Slovak language

The file `slovak.dtx`[32] defines all the language-specific macros for the Slovak language.

For this language the macro `\q` is defined. It is used with the letters (`t`, `d`, `l`, and `L`) and adds a ’ to them to simulate a ‘hook’ that should be there. The result looks like ť.

[31] The file described in this section has version number ? and was last revised on ?.

[32] The file described in this section has version number ? and was last revised on ?. It was written by Jana Chlebikova (`chlebik@euromath.dk`).

32 The Slovanian language

The file `slovene.dtx`[33] defines all the language-specific macros for the Slovanian language.

For this language the character " is made active. In table 14 an overview is given of its purpose. One of the reasons for this is that in the Slovene language some special characters are used.

`"c`	`\"c`, also implemented for the lowercase and uppercase s and z.
`"-`	an explicit hyphen sign, allowing hyphenation in the rest of the word.
`""`	like `"-`, but producing no hyphen sign (for compund words with hyphen, e.g. `x-""y`).
`"‘`	for Slovene left double quotes (looks like ,,).
`"’`	for Slovene right double quotes.
`"<`	for French left double quotes (similar to <<).
`">`	for French right double quotes (similar to >>).

Table 14: The extra definitions made by `slovene.ldf`

33 The Lower Sorbian language

The file `lsorbian.dtx`[34] It defines all the language-specific macros for Lower Sorbian.

34 The Upper Sorbian language

The file `usorbian.dtx`[35] It defines all the language-specific macros for Upper Sorbian.

35 The Turkish language

The file `turkish.dtx`[36] defines all the language definition macros for the Turkish language[37].

Turkish typographic rules specify that a little ‘white space’ should be added before the characters ‘`:`’, ‘`!`’ and ‘`=`’. In order to insert this white space automatically these characters are made ‘active’. Also `\frenhspacing` is set.

[33] The file described in this section has version number ? and was last revised on ?. Contributions were made by Danilo Zavrtanik, University of Ljubljana (YU) and Leon Žlajpah (`leon.zlajpah@ijs.si`).

[34] The file described in this section has version number ? and was last revised on ?. It was written by Eduard Werner (`edi@kaihh.hanse.de`).

[35] The file described in this section has version number ? and was last revised on ?. It was written by Eduard Werner (`edi@kaihh.hanse.de`).

[36] The file described in this section has version number ? and was last revised on ?.

[37] Mustafa Burc, `z6001@rziris01.rrz.uni-hamburg.de` provided the code for this file. It is based on the work by Pierre Mackay

36 The Bahasa language

The file `bahasa.dtx`[38] defines all the language definition macros for the bahasa indonesia / bahasa melayu language. Bahasa just means 'language' in bahasa indonesia / bahasa melayu. Since both national versions of the language use the same writing, although differing in pronounciation, this file can be used for both languages.

For this language currently no special definitions are needed or available.

[38] The file described in this section has version number ? and was last revised on ?.