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

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

Printed December 23, 1999

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 documents can be typeset in a non-american language or in more than one language.

Contents

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

1 The user interface

The user interface of this package is quite simple. It consists of a set of commands that switch from one language to another and a set of commands that deal with shorthands. It is also possible to find out out what the current language is.

`\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.

`otherlanguage` The environment `otherlanguage` 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. The language to switch to is specified as an argument to `\begin{otherlanguage}`.

`\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. This command only switches the extra definitions and the hyphenation rules for the language, *not* the names and dates.

`otherlanguage*` In the environment otherlanguage* only the typesetting is done according to the rules of the other language, but the text-strings such as 'figure', 'table', etc. are left as they were set outside this environment.

`\languagename` The control sequence `\languagename` contains the name of the current language.

`\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` The command `\useshorthands` initiates the definition of user-defined shorthand sequences. It has one argument, the character which starts these 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.

`\aliasshorthand` The command `\aliasshorthand` can be used to let another character perform the same functions as the default shorthand character. If one prefers for example to use the character / over " in typing polish texts this can be acheived by entering `\aliasshorthand{"}{/}`.

`\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.

`\shorthandon`
`\shorthandoff` It is sometimes necessary to be able to switch a shorthand character off temporarily because it needs to be used in an entire different way. For this purpose the user commands `\shorthandoff` and `\shorthandon` are provided. They each take a list of characters as their arguments. The command `\shorthandoff` sets the `\catcode` for each of the characters in its argument to other (12); the command `\shorthandon` sets the `\catcode` to active (13). Both commands only perform their duty on 'known' shorthand characters. If a character is not known to be a shorthand character its category code will be left unchanged.

1.1 Languages supported by Babel

In the following table all the languages supported by Babel are listed, together with the names of the options with which you can load babel for each language.

Language	Option(s)
Afrikaans	afrikaans
Bahasa	bahasa
Breton	breton
Catalan	catalan
Croatian	croatian
Czech	czech
Danish	danish
Dutch	dutch

Language	Option(s)
English	english, USenglish, american, UKenglish, british
Esperanto	esperanto
Estonian	estonian
Finnish	finnish
French	french, francais
Galician	galician
German	austrian, german, germanb, ngerman, naustrian
Greek	greek, polutonikogreek
Hebrew	hebrew
Hungarian	magyar, hungarian
Irish Gaelic	irish
Italian	italian
Lower Sorbian	lowersorbian
Norwegian	norsk, nynorsk
Polish	polish
Portuguese	portuges, portuguese, brazilian, brazil
Romanian	romanian
Russian	russian
Scottish Gaelic	scottish
Spanish	spanish
Slovakian	slovak
Slovenian	slovene
Swedish	swedish
Turkish	turkish
Ukrainian	ukrainian
Upper Sorbian	uppersorbian
Welsh	welsh

For some languages babel supports the options activeacute and activegrave; for typestting russian texts babel knows about the options LWN and LCY to specify the fontencoding of the cyrillic font used. Currently only LWN is supported.

1.2 Workarounds

When you use the document class `book` *and* you use `\ref` inside the argument of `\chapter` you will experience the problem that LaTeX will keep complaining about an undefined label. The reason is that the argument of `\ref` is passed through `\uppercase` at some time during processing. To prevent such problems you could revert to using uppercase labels, or you can use `\lowercase{\ref{foo}}` inside the argument of `\chapter`.

2 Changes for LaTeX 2_ε

With the advent of LaTeX 2_ε 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 LaTeX 2_ε 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 3.6

In Babel version 3.6 a number of bugs that were found in version 3.5 are fixed. Also a number of changes and additions have occured:

- A new environment otherlanguage* is introduced. it only switches the 'specials', but leaves the 'captions' untouched.
- The shorthands are no longer fully expandable. Some problems could only be solved by peeking at the token following an active character. The advantage is that `'{}a` works as expected for languages that have the `'` active.
- Support for typesetting french texts is much enhanced; the file `francais.ldf` is now replaced by `frenchb.ldf` which is maintained by Daniel Flipo.
- Support for typesetting the russian language is again available. The languange definition file was originally developped by Olga Lapko from CyrTUG. The fonts needed to typeset the russian language are now part of the babel distribution. The support is not yet up to the level which is needed according to Olga, but this is a start.
- Support for typesetting greek texts is now also available. What is offered in this release is a first attempt; it will be enhanced later on by Yannis Haralambous.
- in babel 3.6j some hooks have been added for the development of support for Hebrew typesetting.
- Support for typesetting texts in Afrikaans (a variant of Dutch, spoken in South Africa) has been added to `dutch.ldf`.
- Support for typesetting welsh texts is now available.
- A new command `\aliasshorthand` is introduced. It seems that in Poland various conventions are used to type the necessary polish letters. It is now possible to use the character `/` as a shorthand character instead of the character `"` by issuing the command `\aliasshorthand{"}{/}`.

- The shorthand mechanism now deals correctly with characters that are already active.
- Shorthand characters are made active at `\begin{document}`, not earlier. This is to prevent problems with other packages.
- A *preambleonly* command `\substitutefontfamily` has been added to create `.fd` files on the fly when the font families of the latin text differ from the families used for the cyrillic or greek parts of the text.
- Three new comands `\LdfInit`, `\ldf@quit` and `\ldf@finish` are introduced that perform a number of standard tasks.
- In babel 3.6k the language Ukrainian has been added and the support for russian typesetting has been adapted to the package 'cyrillic' to be released with the december 1998 release of LaTeX 2_ε.

4 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:

- the selection of the language is delayed untill `\begin{document}`, this has the consequence that you need to add appropriate `\selectlanguage` commands if you include `\hyphenation` lists in the preamble of your document.
- 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`. It can cantain any valid LaTeX code.

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

In the core of the babel system a number of 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 also be usefull 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. The current format 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*⟩; where ⟨*lang*⟩ is either the name of the language definition file or the name of the LaTeX option that is to be used. 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 language definition 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.

`\langhyphenmins` 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.

`\ProvidesLanguage` The macro `\ProvidesLanguage` should be used to identify the language definition files. It's syntax is similar to the syntax of the LaTeX command `\ProvidesPackage`.

`\LdfInit` The macro `\LdfInit` performs a couple of standard checks that have to be made at the beginning of a language definition file, such as checking the category code of the @-sign, preventing that the `.ldf` file is processed twice, etc.

`\ldf@quit` The macro `\ldf@quit` performs a couple of tasks that need to be taken care of when a `.ldf` file was processed earlier. These tasks include the resetting of the category code of the @-sign, preparing the language to be activated at `\begin{document}` time and ending the input stream.

`\ldf@finish` The macro `\ldf@finish` performs a couple of tasks that need to be taken care of at the end of each `.ldf` file. These tasks include the resetting of the category code of the @-sign, the loading of a local configuration file and preparing the language to be activated at `\begin{document}` time.

`\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. This macro is called by `\ldf@finish`.

`\subsitutefontfamily` This command takes three arguments, a font encoding and two font family names. It creates a font description file for the first font in the given encoding. This `.fd` file will instruct LaTeX to use a font from the second family when a font from the first family in the given encoding seems to be needed.

5.1 Support for active characters

In quite a number of 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`
`\bbl@deactivate` The command `\bbl@activate` is used to change the way an active character expands. `\bbl@activate` 'switches on' the active behaviour of the character. `\bbl@deactivate` 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`
`\bbl@remove@special` The TeXbook states: "Plain TeX includes a macro called `\dospecials` that is essentially a set macro, 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.

5.2 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` 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.

5.3 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`.

5.4 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.

6 Compatibility with `german.sty`

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

[1]This mechanism was introduced by Bernd Raichle.

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

7 Compatibility with ngerman.sty

When used with the options ngerman or naustrian babel will provide all features of the package ngerman. There is however one exception: The commands for special hyphenation of double consonants ("ff etc.) and ck ("ck), which are no longer required with the new German orthography, are undefined. With the ngerman package, however, these commands will generate appropriate warning messages only.

8 Compatibility with the french package

It has been reported to me that the package french by Bernard Gaulle (gaulle@idris.fr) works together with babel.

Therefore, babel will first search for the file french.ldf when you give it the option french; then it will try to load frenchb.ldf. When you give babel the option francais it will only look for frenchb.ldf.

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

9 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.

10 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. 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.

11 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.

`^c`	gives ĉ with hyphenation in the rest of the word allowed, this works for c, C, g, G, H, J, s, S, z, Z
`^h`	prevents ĥ from becoming too tall
`^j`	gives ĵ
`^u`	gives ŭ, with hyphenation in the rest of the word allowed
`^U`	gives Ŭ, with hyphenation in the rest of the word allowed
`^|`	inserts a `\discretionary{-}{}{}`

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

12 The Dutch language

The file `dutch.dtx`[4] defines all the language-specific macros for the Dutch language and the 'Afrikaans' version[5] of it.

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"’`.

13 The English language

The file `english.dtx`[6] 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 v1.4m and was last revised on 1999/04/14. 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 v3.8g, and was last revised on 1999/04/05.

[5]contributed by Stoffel Lombard (`lombc@b31pc87.up.ac.za`)

[6]The file described in this section has version number v3.3i and was last revised on 1999/04/11.

`"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.
`"~`	to produce a hyphencharcter without the following `\discretionary{}{}{}`.
`""`	to produce an invisible 'breakpoint'.
``"` ``	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`

14 The German language

The file `germanb.dtx`[7] defines all the language definition macros for the German language as well as for the Austrian dialect of this language[8].

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

15 The German language – new orthography

The file `ngermanb.dtx`[9] defines all the language definition macros for the German language with the 'new orthography' introduced in August 1998. This includes also the Austrian dialect of this language.

As with the 'traditional' German orthography, the character `"` is made active, and the commands in table 3 can be used, except for `"ck` and `"ff` etc., which are no longer required.

The internal language names are `ngerman` and `naustrian`.

16 The Breton language

The file `breton.dtx`[10] 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

[7]The file described in this section has version number v2.6g and was last revised on 1999/04/05.

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

[9]The file described in this section has version number v2.6h and was last revised on 1999/04/22.

[10]The file described in this section has version number v1.0f and was last revised on 1999/04/21.

`"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`

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.

17 The Welsh language

The file `welsh.dtx`[11] defines all the language definition macros for the Welsh language as well as for the ¡Dialect¿ version of this language.

[11] The file described in this section has version number v1.0b and was last revised on 1999/04/18.

`\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`

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

18 The Irish language

The file `irish.dtx`[12] defines all the language definition macros for the Irish language.

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

19 The Scottish language

The file `scottish.dtx`[13] defines all the language definition macros for the Scottish language.

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

20 The Greek language

The file `greek.dtx`[14] defines all the language definition macros for the Greek language, i.e., as it used today with only one accent, and the *πολυτονικό* ("Polutoniko") Greek dialect for typesetting greek text with all accents. This separation arose out of the need to siplify things, for only very few people will be really interested to typeset multiaccented Greek text.

`\greektext` `\latintext`

The commands `\greektext` and `\latintext` can be used to switch to greek or latin fonts. These are declarations.

`\textgreek` `\textlatin` `\textol`

The commands `\textgreek` and `\textlatin` both take one argument which is then typeset using the requested font encoding. The command `\greekol` switches to the greek outline font family, while the command `\textol` typests a short text in outline font. A number of extra greek characters are made available through the added text commands `\stigma`, `\qoppa`, `\sampi`, `\ddigamma`, `\Digamma`, `\euro`, `\permill`, and `\vardigamma`.

20.1 Typing conventions

Entering greek text can be quite difficult because of the many diacritical signs that need to be added for various purposes. The fonts that are used to typeset Greek make this a lot easier by offering a lot of ligatures. But in order for this to work, some characters need to be considered as letters. These characters are `<`, `>`, `~`, `` ` ``, `'`, `"` and `|`. Therefore their `\lccode` is changed when Greek is in effect. In order to let `\uppercase` give correct results, the `\uccode` of these characters is set to a non-existing character to make them disappear. Of course not all characters are needed when typesetting "modern" *μονοτονικό*. In that case we only need the `'` and `"` symbols which are treated in the proper way.

[12]The file described in this section has version number v1.0g and was last revised on 1999/04/19. A contribution was made by Marion Gunn.

[13]The file described in this section has version number v1.0f and was last revised on 1999/04/19. A contribution was made by Fraser Grant (`FRASER@CERNVM`).

[14]The file described in this section has version number v1.2g and was last revised on 1999/05/17. The original author is Apostolos Syropoulos (`apostolo@platon.ee.duth.gr`), code from `kdgreek.sty` by David Kastrup `dak@neuroinformatik.ruhr-uni-bochum.de` was used to enhance the support for typesetting greek texts.

20.2 Greek numbering

The Greek alphabetical numbering system, like the Roman one, is still used in everyday life for short enumerations. Unfortunately most Greeks don't know how to write Greek numbers bigger than 20 or 30. Nevertheless, in official editions of the last century and beginning of this century this numbering system was also used for dates and numbers in the range of several thousands. Nowadays this numbering system is primary used by the Eastern Orthodox Church and by certain scholars. It is hence necessary to be able to typeset any Greek numeral up to 999 999. Here are the conventions:

- There is no Greek numeral for any number less than or equal to 0.
- Numbers from 1 to 9 are denoted by letters alpha, beta, gamma, delta, epsilon, stigma, zeta, eta, theta, followed by a mark similar to the mathematical symbol "prime". (Nowadays instead of letter stigma the digraph sigma tau is used for number 6. Mainly because the letter stigma is not always available, so people opt to write down the first two letters of its name as an alternative. In our implementation we produce the letter stigma, not the digraph sigma tau.)
- Decades from 10 to 90 are denoted by letters iota, kappa, lambda, mu, nu, xi, omikron, pi, qoppa, again followed by the numeric mark. The qoppa used for this purpose has a special zig-zag form, which doesn't resemble at all the original 'q'-like qoppa.
- Hundreds from 100 to 900 are denoted by letters rho, sigma, tau, upsilon, phi, chi, psi, omega, sampi, followed by the numeric mark.
- Any number between 1 and 999 is obtained by a group of letters denoting the hundreds decades and units, followed by a numeric mark.
- To denote thousands one uses the same method, but this time the mark is placed in front of the letter, and under the baseline (it is inverted by 180 degrees). When a group of letters denoting thousands is followed by a group of letters denoting a number under 1000, then both marks are used.

`\greeknumeral` `\Greeknumeral`

Using these conventions one obtains numbers up to 999 999. The command `\greeknumeral` makes it possible to typeset Greek numerals. There is also an "uppercase" version of this macro: `\Greeknumeral`.

Another system which was in wide use only in Athens, could express any positive number. This system is implemented in package `athnum`.

21 The French language

The file `frenchb.dtx`[15], derived from `frenchy.sty`, defines all the language definition macros for the French language.

Customization for the French language is achieved following the book "Lexique des règles typographiques en usage à l'Imprimerie nationale" troisième édition (1994), ISBN-2-11-081075-0.

[15]The file described in this section has version number v1.3g and was last revised on 1999/08/18.

This file has been designed to be used with LaTeX 2ε, LaTeX-2.09 and PlainTeX formats. If you are still using LaTeX-2.09, you *should* consider switching to LaTeX 2ε!

Any of the commands `\selectlanguage{french}`, `\selectlanguage{francais}`, or `\selectlanguage{frenchb}` switches to the French language with the following effects:

1. French hyphenation patterns are made active;
2. 'double punctuation' is made active for correct spacing in French;
3. `\today` prints the date in French;
4. the caption names are translated into French (LaTeX only);
5. the default items in itemize environment are set to '–' instead of •, and no vertical spacing and no glue is added, a hook to reset standard LaTeX spacing is provided (`\FrenchItemizeSpacingfalse`);
6. vertical spacing in general LaTeX lists is shortened, a hook to reset standard LaTeX settings is provided (`\FrenchListSpacingfalse`) ;
7. the first paragraph of each section is indented (LaTeX only);
8. French quotation marks can be typeset using the commands `\og` and `\fg` which work in LaTeX 2ε, LaTeX-2.09 and PlainTeX, their appearance depending on what is available to draw them; if you use LaTeX 2ε with `T1`-encoding you can also enter them as `<<~French quotation marks~>>` but then *don't forget* the unbreakable spaces, (`\og` and `\fg` provide for correct line breaks);
9. a command `\up` is provided to typeset superscripts like `M\up{me}` (abbreviation for "Madame"), `1\up{er}` (for "premier");
10. family names should be typeset in small capitals and never be hyphenated, the macro `\bsc` (boxed small caps) does this, e.g., `Leslie~\bsc{Lamport}` will produce Leslie LAMPORT;
11. commands `\primo`, `\secundo`, `\tertio` and `\quarto` may be used to enumerate in lists;
12. abbreviations for "Numéro" and "numéro" are obtained via the commands `\No`, `\no`;
13. two commands are provided to typeset abbreviations for "degré": `\degre` prints the raw character and `\degres` should be used to typeset temperatures (e.g., "`20~\degres C`" with an unbreakable space), or for alcohols' strengths (e.g., "`45\degres`" with *no* space in French);
14. a new command `\nombre` is provided to ease the typesetting of numbers: it works both in text and in math-mode: inputting `\nombre{3141,592653}` will format this number properly according to the current language (French or non-French) [16]. The command `\nombre` is a contribution of Vincent Jalby using ideas of David Carlisle in comma.sty.

[16] In math-mode the comma has to be surrounded with braces to avoid a spurious space being inserted after it (see the TeXbook p. 134). Besides this, each slice of three digits should be separated either with a comma in English or with a space in French.

15. `frenchb` has been designed to take advantage of the `xspace` package if present: adding `\usepackage{xspace}` in the preamble will force macros like `\fg`, `\ier`, ..., `\ieme` to respect the spaces you type after them, for instance typing '`1\ier juin`' will print '1$^{\text{er}}$ juin' (no need for a forced space after `1\ier`).

All commands previously available in `francais.ldf` have been included in `frenchb.ldf` for compatibility, sometimes with updated definitions.

The `french` package, by Bernard GAULLE, was not designed to run with babel (although the latest versions claim to be babel compatible), but rather as a standalone package for the French language. It provides many more functionalities (like `\lettrine`, `\sommaire`...) not available in `frenchb`, at the cost of a much greater complexity and possible incompatibilities with other languages.

As `french` is known to produce the best layout available for French typography, I have borrowed many ideas from Bernard's file. I did my best to help users of both packages (`french` and `frenchb`) to exchange their sources files easily, with one exception which affects the way French quotation marks are entered: `frenchb` uses *macros* (`\og` and `\fg`) while `french` uses active characters (`<<` and `>>`).

French typographic rules specify that some white space should be present before 'double punctuation' characters. These characters are `;` `!` `?` and `:`. In order to get this white space automatically, the category code of these characters is made `\active`. In French, the user *should* input these four characters preceded with a space, but as many people forget about it (even among native French writers!), the default behaviour of `frenchb` is to automatically add a `\thinspace` before '`;`' '`!`' '`?`' and a normal (unbreakable) space before '`:`' (this is the rule in French typography). It's up to the user to add or not a space *after* 'double punctuation' characters: usually a space is necessary, but not always (before a full point or a closing brace for instance), so this cannot done automatically.

In (rare) cases where no space should be added before a 'double punctuation', either use `\string;` `\string:` `\string!` `\string?` instead of `;` `:` `!` `?`, or switch locally to `english`. For instance you can type `C\string:TEX` or `\begin{otherlanguage}{english}{C:TEX}\end{otherlanguage}` to avoid the space before `:` in a MS-DOS path.

Some users dislike this automatic insertion of a space before 'double punctuation', and prefer to decide themselves whether a space should be added or not; so a hook `\NoAutoSpaceBeforeFDP` is provided: if this command is added (in file `frenchb.cfg`, or anywhere in a document) `frenchb` will respect your typing, and introduce a suitable space before 'double punctuation' *if and only if* a space is typed in the source file before those signs.

The command `\AutoSpaceBeforeFDP` switches back to the default behaviour of `frenchb`.

Once you have built your format, a good precaution would be to perform some basic tests about hyphenation in French. For LaTeX 2_ε I suggest this:

- run the following file, with the encoding suitable for your machine (*myencoding* will be `latin1` for UNIX machines and PCs running Windows, `applemac` for Macintoshs, or `cp850` for PCs running DOS. If you are using MlTeX together with CM fonts, comment out the line

`\usepackage[`*my-encoding*`]{inputenc}`.

```
%%% Test file for French hyphenation.
\documentclass{article}
\usepackage[my-encoding]{inputenc}
\usepackage[francais]{babel}
\begin{document}
\showhyphens{signal container \'ev\'enement alg\`ebre}
\showhyphens{signal container événement algèbre}
\end{document}
```

- check the hyphenations proposed by TeX in your log-file; in French you should get with both 7-bit and 8-bit encodings
`si-gnal contai-ner évé-ne-ment al-gèbre`.
Do not care about how accented characters are displayed in the log-file, what matters is the position of the '`-`' hyphen signs *only*.

If they are all correct, your installation (probably) works fine, if one (or more) is (are) wrong, ask a local wizard to see what's going wrong and perform the test again (or e-mail me about what happens).

Frequent mismatches:

- you get `sig-nal con-tainer`, this probably means that the hyphenation patterns you are using are for US-english, not for French;
- you get no hyphen at all in `évé-ne-ment`, this probably means that you are using CM fonts and the macro `\accent` to produce accented characters. Consider switching to DC/EC fonts and T1-encoding or use MlTeX.

`frenchb` has been improved using helpful suggestions from many people, the main contributions came from Vincent Jalby. Thanks to all of them!

Changes

First version released: 1.1 as of 1996/05/31 part of babel-3.6beta.

Changes in version 1.1b: update for babel-3.6.

Changes in version 1.2: new command `\nombre` to format numbers; removed command `\fup` borrowed from the `french` package (`\up` does a better job in LaTeX 2_ε); also removed aliases `\french` and `\english` (`frenchb.cfg` is a better place for these).

Changes in version 1.3:

- The 'xspace' package, when present, now controls spacing after all (sensible) macros, formerly 'xspace' worked on `\fg` (suggested by Vincent Jalby);
- spacing after opening and before closing guillemets improved as suggested by Thierry Bouche;
- a replacement for poor looking guillemets in OT1 encoding (*math* fonts are used to emulate them) is provided if the file `ot2wncyr.fd` is found: this file defines an OT2 encoding using AMS Cyrillic fonts; these have built-in guillemets, they are *text* fonts, are free, and are distributed as METAFONT and Type1 fonts (good for pdftex!), the replacement was suggested

by Gérard Degrez; if the files `ot2wncyr.fd`, `wncy*.mf`, `wncy*.tfm`, and `wncy*.pfb` aren't available on your system, you *should* get them from CTAN; if your system complains about missing `wncy*.tfm` fonts, it means your TeX system is *incomplete*, as a quick fix, you can either remove `ot2wncyr.fd` or add the command `LasyGuillemets` to the preamble of your document or to `frenchb.cfg`, then `frenchb` will behave as it did in the previous versions.

- environment 'itemize' has been redesigned in French according to specifications from Jacques André and Thierry Bouche;
- two switches have been added to go back to standard LaTeX list spacing `\FrenchItemizeSpacingfalse` and `\FrenchListSpacingfalse`;
- `\nombre` now properly handles signs in LaTeX 2_ε;
- definition of `\dots` changed in French;
- in French, captions in figures and tables are printed with an endash instead of a colon.

22 The Italian language

The file `italian.dtx`[17] It defines all the language-specific macros for the Italian language.

For this language the `\clubpenalty`, `\widowpenalty` are set to rather high values and `\finalhyphendemerits` is set to such a high value that hyphenation is prohibited between the last two lines of a paragraph.

Claudio Beccari (`beccari@polito.it`) added some specific extras for typesetting units according to ISO 31/XI and apices and pedices (superscripts and subscripts) for both text and math, in the latter case allowing correct application of ISO 31/XI.

23 The Portuguese language

The file `portuges.dtx`[18] 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.

24 The Spanish language

The file `spanish.dtx`[19] defines all the language definition macro's for the Spanish[20] language.

[17] The file described in this section has version number v1.2l and was last revised on 1999/05/01. The original author is Maurizio Codogno, (`mau@beatles.cselt.stet.it`). It has largely been revised by Johannes Braams and Claudio Beccari.

[18] The file described in this section has version number v1.2l and was last revised on 1999/04/16. Contributions were made by Jose Pedro Ramalhete (`JRAMALHE@CERNVM` or `Jose-Pedro_Ramalhete@MACMAIL`) and Arnaldo Viegas de Lima `arnaldo@VNET.IBM.COM`.

[19] The file described in this section has version number v3.4i and was last revised on 1999/04/19. The original author is Julio Sánchez, (`jsanchez@gmv.es`).

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

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

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

This file[21] incorporates the result of discussions held in the Spanish-TEX[22] electronic mail list.

For this language the characters `~` and `"` are used as shorthand characters. In table 6 an overview is given of their purpose.

<table>
<tr><td><code>"|</code></td><td>disable ligature at this position.</td></tr>
<tr><td><code>"-</code></td><td>an explicit hyphen sign, allowing hyphenation in the rest of the word.</td></tr>
<tr><td><code>""</code></td><td>like <code>"-</code>, but producing no hyphen sign (for words that should break at some sign such as “entrada/salida.”</td></tr>
<tr><td><code>\-</code></td><td>like the old <code>\-</code>, but allowing hyphenation in the rest of the word.</td></tr>
<tr><td><code>"u</code></td><td>a u with dieresis allowing hyphenation.</td></tr>
<tr><td><code>"a</code></td><td>feminine ordinal as in 1ª.</td></tr>
<tr><td><code>"o</code></td><td>masculine ordinal as in 1º.</td></tr>
<tr><td><code>"<</code></td><td>for French left double quotes (similar to <<).</td></tr>
<tr><td><code>"></code></td><td>for French right double quotes (similar to >>).</td></tr>
<tr><td><code>~n</code></td><td>a n with tilde. Works for uppercase too.</td></tr>
</table>

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

When the user supplies `babel` with the option activeacute the character `'` will also become a shorthand character. It's use is indicated in table 7. All shorthand characters behave according to their original definitions if not followed by one of the characters indicated in the table(s).

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

[21] 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.

[22] `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`.

`'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.

Table 7: The extra definitions made by `spanish.ldf`

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[23] 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.

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.

25 The Catalan language

The file `catalan.dtx`[24] 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 8 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 9 also become available[25].

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

[23] 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.

[24] The file described in this section has version number v2.2k and was last revised on 1999/05/05.

[25] 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 to type `l'{}estri` instead of `l'estri`.

`\l.l`	geminated-l digraph (similar to l·l). `\L.L` produces the uppercase version.
`\lgem`	geminated-l digraph (similar to l·l). `\Lgem` produces the uppercase version.
`\up`	Macro to help typing raised ordinals, like 1$^{\text{er}}$. 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 8: Extra definitions made by file `catalan.ldf` (activated by default)

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

26 The Galician language

The file `galician.dtx`[26] defines all the language definition macros for the Galician language.

For this language the characters `'` `~` and `"` are made active. In table 10 an overview is given of their purpose. These active accents character behave according to their original definitions if not followed by one of the characters indicated in that table.

[26]The file described in this section has version number v1.2h and was last revised on 1999/04/21.

- `"|` 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 $1^{\underline{a}}$.
- `"o` masculine ordinal as in $1^{\underline{o}}$.
- `~n` a n with tilde. Works for uppercase too.

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

27 The Romanian language

The file `romanian.dtx`[27] defines all the language-specific macros for the Romanian language.

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

28 The Danish language

The file `danish.dtx`[28] defines all the language definition macros for the Danish language.

For this language the character `"` is made active. In table 11 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 11: The extra definitions made by `danish.ldf`

[27]The file described in this section has version number v1.2j and was last revised on 1999/04/16. A contribution was made by Umstatter Horst (`hhu@cernvm.cern.ch`).

[28]The file described in this section has version number v1.3l and was last revised on 1999/04/11. A contribution was made by Henning Larsen (`larsen@cernvm.cern.ch`)

29 The Norwegian language

The file `norsk.dtx`[29] 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.

30 The Swedish language

The file `swedish.dtx`[30] defines all the language-specific macros for the Swedish language.

For this language the character `"` is made active. In table 12 an overview is given of its purpose. The vertical placement of the "umlaut" in some letters can be controlled this way.

`"a`	`\"a`, also implemented for A, o and O.
`"w`	gives å, also works for uppercase letters.
`"ff`	for `ff` to be hyphenated as `ff-f`, this is also implemented for b, d, f, g, l, m, n, p, r, s, and t.
`"|`	disable ligature at this position.
`"-`	an explicit hyphen sign, allowing hyphenation in the rest of the word.
`""`	like `"-`, but producing no hyphen sign (for compound 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.

Table 12: The extra definitions made by `swedish.sty`

31 The Finnish language

The file `finnish.dtx`[31] defines all the language definition macros for the Finnish language.

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

[29] The file described in this section has version number v1.2j and was last revised on 1999/04/22. 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`).

[30] The file described in this section has version number v2.2 and was last revised on 1999/04/18. Contributions were made by Sten Hellman (`HELLMAN@CERNVM.CERN.CH`) and Erik Öshols (`erik@ine.kfk.de`).

[31] The file described in this section has version number v1.3m and was last revised on 1999/04/13. 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 13: The extra definitions made by `finnish.ldf`

32 The Hungarian language

The file option `magyar.dtx`[32] 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'.

33 The Estonian language

The file `estonian.dtx`[33] 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 14. 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

[32] The file described in this section has version number v1.3j and was last revised on 1999/04/22. 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`).

[33] The file described in this section has version number v1.0g and was last revised on 1999/04/11. 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 14: 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.

34 The Croatian language

The file croatian.dtx[34] defines all the language definition macros for the Croatian language.

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

35 The Czech language

The file czech.dtx[35] 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.

[34]The file described in this section has version number v1.3j and was last revised on 1999/03/12. A contribution was made by Alan Paić (paica@cernvm.cern.ch).

[35]The file described in this section has version number v1.3i and was last revised on 1999/04/11. Contributions were made by Milos Lokajicek (LOKAJICK@CERNVM).

36 The Polish language

The file `polish.dtx`[36] defines all the language-specific macros for the Polish language.

For this language the character `"` is made active. In table 15 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 15: The extra definitions made by `polish.sty`

37 The Slovak language

The file `slovak.dtx`[37] 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 t'.

[36] The file described in this section has version number v1.2g and was last revised on 1999/04/13.

[37] The file described in this section has version number v1.2k and was last revised on 1999/04/18. It was written by Jana Chlebikova (`chlebik@euromath.dk`).

38 The Slovenian language

The file `slovene.dtx`[38] defines all the language-specific macros for the Slovenian language.

For this language the character `"` is made active. In table 16 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 16: The extra definitions made by `slovene.ldf`

39 The Russian language

The file `russianb.dtx`[39] defines all the language-specific macros for the Russian language. It needs the file `cyrcod` for success documentation with Russian encodings (see below).

For this language the character `"` is made active. In table 17 an overview is given of its purpose.

The quotes in table 17 can also be typeset by using the commands in table 18.

The French quotes are also available as ligatures ‘`<<`’ and ‘`>>`’ in 8-bit Cyrillic font encodings (`LCY`, `X2`, `T2*`) and as ‘`<`’ and ‘`>`’ characters in 7-bit Cyrillic font encodings (`OT2` and `LWN`).

The quotation marks traditionally used in Russian were borrowed from other languages (e.g., French and German) so they keep their original names.

40 The Ukrainian language

The file `ukraineb.dtx`[40] defines all the language-specific macros for the Ukrainian language. It needs the file `cyrcod` for success documentation with Ukrainian encodings (see below).

[38] The file described in this section has version number v1.2k and was last revised on 1999/04/19. Contributions were made by Danilo Zavrtanik, University of Ljubljana (YU) and Leon Žlajpah (`leon.zlajpah@ijs.si`).

[39] 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 Russian. Later the definitions from `russian.sty` version 1.0b (for LaTeX 2.09), `russian.sty` version v2.5c (for LaTeX 2_ε) and `francais.sty` version 4.5c and `germanb.sty` version 2.5c were added.

[40] The file described in this section has version number ?. This file was derived from the `russianb.dtx` version 1.1g.

<table>
<tr><td>"|</td><td>disable ligature at this position.</td></tr>
<tr><td>"-</td><td>an explicit hyphen sign, allowing hyphenation in the rest of the word.</td></tr>
<tr><td>"---</td><td>Cyrillic emdash in plain text.</td></tr>
<tr><td>"--~</td><td>Cyrillic emdash in compound names (surnames).</td></tr>
<tr><td>"--*</td><td>Cyrillic emdash for denoting direct speech.</td></tr>
<tr><td>""</td><td>like "-, but producing no hyphen sign (for compund words with hyphen, e.g. x-""y or some other signs as “disable/enable”).</td></tr>
<tr><td>"~</td><td>for a compound word mark without a breakpoint.</td></tr>
<tr><td>"=</td><td>for a compound word mark with a breakpoint, allowing hyphenation in the composing words.</td></tr>
<tr><td>",</td><td>thinspace for initials with a breakpoint in following surname.</td></tr>
<tr><td>"`</td><td>for German left double quotes (looks like „).</td></tr>
<tr><td>"'</td><td>for German right double quotes (looks like “).</td></tr>
<tr><td>” "<</td><td>for French left double quotes (looks like ≪).</td></tr>
<tr><td>"></td><td>for French right double quotes (looks like ≫).</td></tr>
</table>

Table 17: The extra definitions made by russianb

\cdash---	Cyrillic emdash in plain text.
\cdash--~	Cyrillic emdash in compound names (surnames).
\cdash--*	Cyrillic emdash for denoting direct speech.
\glqq	for German left double quotes (looks like „).
\grqq	for German right double quotes (looks like “).
” \flqq	for French left double quotes (looks like ≪).
\frqq	for French right double quotes (looks like ≫).
\dq	the original quotes character (").

Table 18: More commands which produce quotes, defined by babel

For this language the character " is made active. In table 17 an overview is given of its purpose.

The quotes in table 19 (see, also table 17) can also be typeset by using the commands in table 20 (see, also table 18).

The French quotes are also available as ligatures ‘<<’ and ‘>>’ in 8-bit Cyrillic font encodings (LCY, X2, T2*) and as ‘<’ and ‘>’ characters in 7-bit Cyrillic font encodings (OT2 and LWN).

The quotation marks traditionally used in Ukrainian and Russian languages were borrowed from other languages (e.g. French and German) so they keep their original names.

`"|`	disable ligature at this position.
`"-`	an explicit hyphen sign, allowing hyphenation in the rest of the word.
`"---`	Cyrillic emdash in plain text.
`"--~`	Cyrillic emdash in compound names (surnames).
`"--*`	Cyrillic emdash for denoting direct speech.
`""`	like `"-`, but producing no hyphen sign (for compund words with hyphen, e.g. `x-""y` or some other signs as “disable/enable”).
`"~`	for a compound word mark without a breakpoint.
`"=`	for a compound word mark with a breakpoint, allowing hyphenation in the composing words.
`",`	thinspace for initials with a breakpoint in following surname.
`"‘`	for German left double quotes (looks like „).
`"’`	for German right double quotes (looks like “).
” `"<`	for French left double quotes (looks like $\ll$).
`">`	for French right double quotes (looks like $\gg$).

Table 19: The extra definitions made by `ukraineb`

`\cdash---`	Cyrillic emdash in plain text.
`\cdash--~`	Cyrillic emdash in compound names (surnames).
`\cdash--*`	Cyrillic emdash for denoting direct speech.
`\glqq`	for German left double quotes (looks like „).
`\grqq`	for German right double quotes (looks like “).
” `\flqq`	for French left double quotes (looks like $\ll$).
`\frqq`	for French right double quotes (looks like $\gg$).
`\dq`	the original quotes character (").

Table 20: More commands which produce quotes, defined by babel

41 The Lower Sorbian language

The file `lsorbian.dtx`[41] It defines all the language-specific macros for Lower Sorbian.

42 The Upper Sorbian language

The file `usorbian.dtx`[42] It defines all the language-specific macros for Upper Sorbian.

[41]The file described in this section has version number v1.0e and was last revised on 1999/04/16. It was written by Eduard Werner (`edi@kaihh.hanse.de`).

[42]The file described in this section has version number v1.0g and was last revised on 1998/06/07. It was written by Eduard Werner (`edi@kaihh.hanse.de`).

43 The Turkish language

The file `turkish.dtx`[43] defines all the language definition macros for the Turkish language[44].

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.

44 The Bahasa language

The file `bahasa.dtx`[45] 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.

[43]The file described in this section has version number v1.2l and was last revised on 1999/05/17.

[44]Mustafa Burc, `z6001@rziris01.rrz.uni-hamburg.de` provided the code for this file. It is based on the work by Pierre Mackay; Turgut Uyar, `uyar@cs.itu.edu.tr` supplied additional translations in version 1.2j and later

[45]The file described in this section has version number v1.0g and was last revised on 1999/04/23.