

# A Babel language definition file for French

Daniel FLIPO  
daniel.flipo@free.fr

## 1 The French language

The file `frenchb.dtx`<sup>1</sup>, defines all the language definition macros for the French language.

Customisation 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.

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

`frenchb` has been improved using helpful suggestions from many people, mainly from Jacques André, Michel Bovani, Thierry Bouche, and Vincent Jalby. Thanks to all of them!

This new version (2.x) has been designed to be used with L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> and PlainT<sub>E</sub>X formats only. L<sup>A</sup>T<sub>E</sub>X-2.09 is no longer supported. Changes between version 1.6 and v2.6b are listed in subsection 1.4 p. 6.

An extensive documentation is available in French here:

<http://daniel.flipo.free.fr/frenchb>

### 1.1 Basic interface

In a multilingual document, some typographic rules are language dependent, i.e. spaces before ‘double punctuation’ (: ; ! ?) in French, others concern the general layout (i.e. layout of lists, footnotes, indentation of first paragraphs of sections) and should apply to the whole document.

Starting with version 2.2, `frenchb` behaves differently according to `babel`’s *main language* defined as the *last* option<sup>2</sup> at `babel`’s loading. When French is not `babel`’s main language, `frenchb` no longer alters the global layout of the document (even in parts where French is the current language): the layout of lists, footnotes, indentation of first paragraphs of sections are not customised by `frenchb`.

When French is loaded as the last option of `babel`, `frenchb` makes the following changes to the global layout, *both in French and in all other languages*<sup>3</sup>:

1. the first paragraph of each section is indented (L<sup>A</sup>T<sub>E</sub>X only);
2. the default items in itemize environment are set to ‘—’ instead of ‘•’, and all vertical spacing and glue is deleted; it is possible to change ‘—’ to something else (‘-’ for instance) using `\frenchbsetup{}`;
3. vertical spacing in general L<sup>A</sup>T<sub>E</sub>X lists is shortened;
4. footnotes are displayed “à la française”.

---

<sup>1</sup>The file described in this section has version number v2.6b and was last revised on 2013/03/14.

<sup>2</sup>Its name is kept in `\bb1@main@language`.

<sup>3</sup>For each item, hooks are provided to reset standard L<sup>A</sup>T<sub>E</sub>X settings or to emulate the behavior of former versions of `frenchb` (see command `\frenchbsetup{}`, section 1.2).

Regarding local typography, the command `\selectlanguage{french}` switches to the French language<sup>4</sup>, with the following effects:

1. French hyphenation patterns are made active;
2. ‘double punctuation’ (: ; ! ?) is made active<sup>5</sup> for correct spacing in French;
3. `\today` prints the date in French;
4. the caption names are translated into French (L<sup>A</sup>T<sub>E</sub>X only);
5. the space after `\dots` is removed in French.

Some commands are provided in `frenchb` to make typesetting easier:

1. French quotation marks can be entered using the commands `\og` and `\fg` which work in L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> and PlainT<sub>E</sub>X, their appearance depending on what is available to draw them; even if you use L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> and T1-encoding, you should refrain from entering them as `<<~French quotation marks~>>`: `\og` and `\fg` provide better horizontal spacing. `\og` and `\fg` can be used outside French, they typeset then English quotes “ and ”.
2. A command `\up` is provided to typeset superscripts like `M\up{me}` (abbreviation for “Madame”), `1\up{er}` (for “premier”). Other commands are also provided for ordinals: `\ier`, `\iere`, `\iers`, `\ieres`, `\ieme`, `\iemes` (`3\iemes` prints 3<sup>es</sup>).
3. 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. Note that composed names (such as Dupont-Durant) may now be hyphenated on explicit hyphens, this differs from `frenchb` v. 1.x.
4. Commands `\primo`, `\secundo`, `\tertio` and `\quarto` print 1<sup>o</sup>, 2<sup>o</sup>, 3<sup>o</sup>, 4<sup>o</sup>. `\FrenchEnumerate{6}` prints 6<sup>o</sup>.
5. Abbreviations for “Numéro(s)” and “numéro(s)” (N<sup>o</sup> N<sup>os</sup> n<sup>o</sup> and n<sup>os</sup> ) are obtained via the commands `\No`, `\Nos`, `\no`, `\nos`.
6. Two commands are provided to typeset the symbol 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).
7. In math mode the comma has to be surrounded with braces to avoid a spurious space being inserted after it, in decimal numbers for instance (see the T<sub>E</sub>Xbook p. 134). The command `\DecimalMathComma` makes the comma be an ordinary character *in French only* (no space added); as a counterpart, if `\DecimalMathComma` is active, an explicit space has to be added in lists and intervals: `$[0,\ 1]$, $(x,\ y)$`. `\StandardMathComma` switches back to the standard behaviour of the comma.

---

<sup>4</sup>`\selectlanguage{français}` and `\selectlanguage{frenchb}` are kept for backward compatibility but should no longer be used.

<sup>5</sup>Actually, they are active in the whole document, only their expansions differ in French and outside French.

8. A command `\nombre` was provided in 1.x versions to easily format numbers in slices of three digits separated either by a comma in English or with a space in French; `\nombre` is now mapped to `\numprint` from `numprint.sty`, see `numprint.pdf` for more information.
9. `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`, `\dots`, ..., to respect the spaces you type after them, for instance typing ‘`1\ier juin`’ will print ‘1<sup>er</sup> juin’ (no need for a forced space after `1\ier`).

## 1.2 Customisation

Up to version 1.6, customisation of `frenchb` was achieved by entering commands in `frenchb.cfg`. This possibility remains for compatibility, but *should not longer be used*. Version 2.0 introduces a new command `\frenchbsetup{}` using the `keyval` syntax which should make it easier to choose among the many options available. The command `\frenchbsetup{}` is to appear in the preamble only (after loading `babel`).

`\frenchbsetup{ShowOptions}` prints all available options to the `.log` file, it is just meant as a remainder of the list of offered options. As usual with `keyval` syntax, boolean options (as `ShowOptions`) can be entered as `ShowOptions=true` or just `ShowOptions`, the ‘`=true`’ part can be omitted.

The other options are listed below. Their default value is shown between brackets, sometimes followed by a ‘`*`’. The ‘`*`’ means that the default shown applies when `frenchb` is loaded as the *last* option of `babel` —`babel`’s *main language*—, and is toggled otherwise:

- `StandardLayout=true` [`false*`] forces `frenchb` not to interfere with the layout: no action on any kind of lists, first paragraphs of sections are not indented (as in English), no action on footnotes. This option replaces the former command `\StandardLayout`. It might be used to avoid conflicts with classes or packages which customise lists or footnotes.
- `GlobalLayoutFrench=false` [`true*`] can be used, when French is the main language, to emulate what prior versions of `frenchb` (pre-2.2) did: lists, and first paragraphs of sections will be displayed the standard way in other languages than French, and “à la française” in French. Note that the layout of footnotes is language independent anyway (see below `FrenchFootnotes` and `AutoSpaceFootnotes`). This option replaces the former command `\FrenchLayout`.
- `ReduceListSpacing=false` [`true*`]; `frenchb` reduces the values of the vertical spaces used in the *all* list environments in French (this includes `itemize`, `enumerate`, `description`, but also `abstract`, `quote`, `quotation` and `verse` and possibly others). Setting this option to `false` reverts to the standard settings of `list`. This option replaces the former command `\FrenchListSpacingfalse`.
- `ListOldLayout=true` [`false`]; starting with version 2.6a, the layout of lists has changed regarding leftmargins’ sizes and default `itemize` label (‘—’ in-

stead of ‘–’ up to 2.5k). This option, provided for backward compatibility, displays lists as they were up to version 2.5k.

- `CompactItemize=false` [`true*`]; should no longer be used (kept only for backward compatibility).
- `StandardItemizeEnv=true` [`false*`]; `frenchb` redefines the `itemize` environment to suppress any vertical space between items of `itemize` lists in French and customize leftmargins. Setting this option to `false` reverts to the standard definition of `itemize`.
- `StandardEnumerateEnv=true` [`false*`]; starting with version 2.6a `frenchb` redefines the `enumerate` environment to make leftmargins match those of the French version of `itemize` lists. Setting this option to `false` reverts to the standard definition of `enumerate`.
- `StandardItemLabels=true` [`false*`] when set to `true` this option stops `frenchb` from changing the labels in `itemize` lists in French.
- `ItemLabels=\textendash, \textbullet, \ding{43}, \dots, [\textemdash*]`; when `StandardItemLabels=false` (the default), this option enables to choose the label used in `itemize` lists for all levels. The next three options do the same but each one for one level only. Note that the example `\ding{43}` requires `\usepackage{pifont}`.
- `ItemLabeli=\textendash, \textbullet, \ding{43}, \dots, [\textemdash*]`
- `ItemLabelii=\textendash, \textbullet, \ding{43}, \dots, [\textemdash*]`
- `ItemLabeliii=\textendash, \textbullet, \ding{43}, \dots, [\textemdash*]`
- `ItemLabeliv=\textendash, \textbullet, \ding{43}, \dots, [\textemdash*]`
- `StandardLists=true` [`false*`] forbids `frenchb` to customise any kind of list. Try the option `StandardLists` in case of conflicts with classes or packages that customise lists too. This option is just a shorthand for all four options `ReduceListSpacing=false`, `StandardItemizeEnv=true`, `StandardEnumerateEnv=true` and `StandardItemLabels=true`.
- `IndentFirst=false` [`true*`]; `frenchb` normally forces indentation of the first paragraph of sections. When this option is set to `false`, the first paragraph of will look the same in French and in English (not indented).
- `FrenchFootnotes=false` [`true*`] reverts to the standard layout of footnotes. By default `frenchb` typesets leading numbers as ‘1.’ instead of ‘1’, but has no effect on footnotes numbered with symbols (as in the `\thanks` command). The former commands `\StandardFootnotes` and `\FrenchFootnotes` are still there, `\StandardFootnotes` can be useful when some footnotes are numbered with letters (inside minipages for instance).
- `AutoSpaceFootnotes=false` [`true*`] ; by default `frenchb` adds a thin space in the running text before the number or symbol calling the footnote. Making this option `false` reverts to the standard setting (no space added).

- `FrenchSuperscripts=false [true]` ; then `\up=\textsuperscript` (option added in version 2.1). Should only be made `false` to recompile older documents. By default `\up` now relies on `\fup` designed to produce better looking superscripts.
- `AutoSpacePunctuation=false [true]`; in French, the user *should* input a space before the four characters ‘:;!?’ 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 recommended by the French Imprimerie nationale). This is convenient in most cases but can lead to addition of spurious spaces in URLs or in MS-DOS paths but only if they are not typed using `\texttt` or verbatim mode. When the current font is a monospaced (typewriter) font, `AutoSpacePunctuation` is locally switched to `false`, no spurious space is added in that case, so the default behaviour of `frenchb` in that area should be fine in most circumstances.

Choosing `AutoSpacePunctuation=false` will ensure that a proper space will be added before ‘:;!?’ *if and only if* a (normal) space has been typed in. Those who are unsure about their typing in this area should stick to the default option and type `\string; \string: \string! \string?` instead of `; : ! ?` in case an unwanted space is added by `frenchb`.

- `ThinColonSpace=true [false]` changes the normal (unbreakable) space added before the colon ‘:’ to a thin space, so that the same amount of space is added before any of the four double punctuation characters. The default setting is supported by the French Imprimerie nationale.
- `LowercaseSuperscripts=false [true]` ; by default `frenchb` inhibits the uppercasing of superscripts (for instance when they are moved to page headers). Making this option `false` will disable this behaviour (not recommended).
- `PartNameFull=false [true]`; when true, `frenchb` numbers the title of `\part{}` commands as “Première partie”, “Deuxième partie” and so on. With some classes which change the `\part{}` command (AMS and SMF classes do so), you will get “Première partie I”, “Deuxième partie II” instead; when this occurs, this option should be set to `false`, part titles will then be printed as “Partie I”, “Partie II”.
- `SuppressWarning=true [false]`; when true `frenchb` issues no warnings if `\@makecaption` has been redefined or if the `bigfoot` package is in use.
- `og=«, fg=»`; when guillemets characters are available on the keyboard (through a compose key for instance), it is nice to use them instead of typing `\og` and `\fg`. This option tells `frenchb` which characters are opening and closing French guillemets (they depend on the input encoding), then you can type either « `guillemets` », or «`guillemets`» (with or without spaces), to get properly typeset French quotes. This option requires `inputenc` to be loaded with the proper encoding, it works with 8-bits encodings (latin1, latin9, ansinew, applemac,...) and multi-byte encodings (utf8 and utf8x).

### 1.3 Hyphenation checks

Once you have built your format, a good precaution would be to perform some basic tests about hyphenation in French. For L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> I suggest this:

- run the following file, with the encoding suitable for your machine (*my-encoding* will be `latin1` for UNIX machines, `ansinew` for PCs running Windows, `applemac` or `latin1` for Macintoshes, or `utf8`...

```
%% Test file for French hyphenation.
\documentclass{article}
\usepackage[my-encoding]{inputenc}
\usepackage[T1]{fontenc} % Use LM fonts
\usepackage{lmodern}      % for French
\usepackage[frenchb]{babel}
\begin{document}
\showhyphens{signal container \’ev\’enement alg\’ebre}
\showhyphens{signal container événement algèbre}
\end{document}
```

- check the hyphenations proposed by T<sub>E</sub>X 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. Using 8-bits fonts with built-in accented characters avoids this kind of mismatch.

**Options’ order** – Please remember that options are read in the order they appear inside the `\frenchbsetup` command. Someone wishing that `frenchb` leaves the layout of lists and footnotes untouched but caring for indentation of first paragraph of sections could choose `\frenchbsetup{StandardLayout,IndentFirst}` and get the expected layout. Choosing `\frenchbsetup{IndentFirst,StandardLayout}` would not lead to the expected result: option `IndentFirst` would be overwritten by `StandardLayout`.

### 1.4 Changes

#### What’s new in version 2.0?

Here is the list of all changes:

- Support for L<sup>A</sup>T<sub>E</sub>X-2.09 and for L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> in compatibility mode has been dropped. This version is meant for L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> and Plain based formats (like **bplain**). L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> formats based on m<sup>L</sup>T<sub>E</sub>X are no longer supported either (plenty of good 8-bits fonts are available now, so T1 encoding should be preferred for typesetting in French). A warning is issued when OT1 encoding is in use at the `\begin{document}`.
- Customisation should now be handled by command `\frenchbsetup{}`, **frenchb.cfg** (kept for compatibility) should no longer be used. See section 1.2 for the list of available options.
- Captions in figures and table have changed in French: former abbreviations “Fig.” and “Tab.” have been replaced by full names “Figure” and “Table”. If this leads to formatting problems in captions, you can add the following two commands to your preamble (after loading **babel**) to get the former captions
 

```
\addto\captionsfrench{\def\figurename{{\scshape Fig.}}}
\addto\captionsfrench{\def\tablename{{\scshape Tab.}}}
```
- The `\nombre` command is now provided by the **numprint** package which has to be loaded *after* **babel** with the option **autolanguage** if number formatting should depend on the current language.
- The `\bsc` command no longer uses an `\hbox` to stop hyphenation of names but a `\kern0pt` instead. This change enables **microtype** to fine tune the length of the argument of `\bsc`; as a side-effect, compound names like Dupont-Durand can now be hyphenated on explicit hyphens. You can get back to the former behaviour of `\bsc` by adding
 

```
\renewcommand*{\bsc}[1]{\leavevmode\hbox{\scshape #1}}
```

 to the preamble of your document.
- Footnotes are now displayed “à la française” for the whole document, except with an explicit
 

```
\frenchbsetup{AutoSpaceFootnotes=false,FrenchFootnotes=false}.
```

 Add this command if you want standard footnotes. It is still possible to revert locally to the standard layout of footnotes by adding `\StandardFootnotes` (inside a **minipage** environment for instance).

### What’s new in version 2.1?

New command `\fup` to typeset better looking superscripts. Former command `\up` is now defined as `\fup`, an option `\frenchbsetup{FrenchSuperscripts=false}` is provided for backward compatibility. `\fup` was designed using ideas from Jacques André, Thierry Bouche and René Fritz, thanks to them!

### What’s new in version 2.2?

Starting with version 2.2a, **frenchb** alters the layout of lists, footnotes, and the indentation of first paragraphs of sections) *only if* French is the “main language” (i.e. **babel**’s last language option). The layout is global for the whole document: lists, etc. look the same in French and in other languages, everything is typeset “à la française” if French is the “main language”, otherwise **frenchb** doesn’t change anything regarding lists, footnotes, and indentation of paragraphs.

### What’s new in version 2.3?

Starting with version 2.3a, **frenchb** no longer inserts spaces automatically before ‘:;!?’ when a typewriter font is in use; this was suggested by Yannis Haralambous to prevent spurious spaces in computer source code or expressions like `C:/foo`, `http://foo.bar`, etc. An option (`OriginalTypewriter`) is provided to get back to the former behaviour of **frenchb**.

Another probably invisible change: lowercase conversion in `\up{}` is now achieved by the  $\text{\LaTeX}$  command `\MakeLowercase` instead of  $\text{\TeX}$ ’s `\lowercase` command. This prevents error messages when diacritics are used inside `\up{}` (diacritics should *never* be used in superscripts though!).

### What’s new in version 2.4?

A new option `SuppressWarning` has been added (desactivated by default) to suppress warnings if `\@makecaption` has been redefined or if the **bigfoot** package is in use.

French hyphenation patterns are now coded in Unicode, see file `hyph-fr.tex`. Extra code has been added to deal with hyphenation of the French “apostrophe” with  $\text{\XeTeX}$  and  $\text{\LuaTeX}$  engines.

Better compatibility with the **enumitem** package.

When typewriter fonts are in use (hence in verbatim mode) no space is added after ‘«’ and before ‘»’ when they are entered as characters (see `\frenchbsetup`).

### What’s new in version 2.5?

The main change is that active characters are no longer used in French with (recent)  $\text{\XeTeX}$ -based engines (they still are with  $\text{\TeX}$ -based engines). All the functionalities (automatic insertion of missing spaces before :;!? or bare replacement of typed spaces with suitable unbreachable ones, tuning of the spaces width) remain available and the user interface is unchanged. The use of active characters is replaced by the `\XeTeXinterchartoks` mechanism (as in package **polyglossia**). A new command `\NoAutoSpacing` has been added. It should be used *inside a group* instead of `\shorthandoff{;:!?}` whenever active characters or automatic spacing of French punctuation or quote characters conflict with other packages; it is designed to work with  $\text{\TeX}$ - and  $\text{\XeTeX}$ -based engines.

Bug corrections: `\frenchspacing` and `\nonfrenchspacing` are no longer messed up by `frenchb.ldf`.

### What’s new in version 2.6?

The way **frenchb** handles list environments has been completely redesigned in version 2.6a due to a long standing bug affecting enumerate lists inside itemize lists. Horizontal indentation of itemize and enumerate lists differs now from previous versions, an option for backward compatibility is provided: `\frenchbsetup{ListOldLayout}`.

**frenchb** is now compatible with the **paralist** package.



## 1.5 File frenchb.cfg

frenchb.cfg is now a dummy file just kept for compatibility with previous versions.

```
1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 %%%%%%%%% WARNING: THIS FILE SHOULD NO LONGER BE USED %%%%%%%%%
3 %% If you want to customise frenchb, please DO NOT hack into the code!
4 %% Do not put any code in this file either, please use the new command
5 %% \frenchbsetup{} with the proper options to customise frenchb.
6 %%
7 %% Add \frenchbsetup{ShowOptions} to your preamble to see the list of
8 %% available options and/or read the documentation.
9 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

## 1.6 Initial setup

While this file was read through the option frenchb we make it behave as if french was specified.

```
10 \def\CurrentOption{french}
```

The macro \LdfInit takes care of preventing that this file is loaded more than once, checking the category code of the @ sign, etc.

```
11 \LdfInit\CurrentOption\datefrench
```

```
\ifLaTeXe No support is provided for late LATEX-2.09: issue a warning and exit if LATEX-2.09
is in use. Plain is still supported.
```

```
12 \newif\ifLaTeXe
13 \let\bbl@tempa\relax
14 \ifx\magnification\@undefined
15   \ifx\@compatibilitytrue\@undefined
16     \PackageError{frenchb.ldf}
17       {LaTeX-2.09 format is no longer supported.\MessageBreak
18       Aborting here}
19     {Please upgrade to LaTeX2e!}
20     \let\bbl@tempa\endinput
21   \else
22     \LaTeXetrue
23   \fi
24 \fi
25 \bbl@tempa
```

Check if hyphenation patterns for the French language have been loaded in language.dat; we allow for the names ‘french’, ‘français’, ‘canadien’ or ‘acadian’. The latter two are both names used in Canada for variants of French that are in use in that country.

```
26 \ifx\l@french\@undefined
27   \ifx\l@français\@undefined
28     \ifx\l@canadien\@undefined
29       \ifx\l@acadian\@undefined
30         \@nopatterns{French}
31         \adddialect\l@french0
32       \else
```

```

33      \let\l@french\l@acadian
34      \fi
35    \else
36      \let\l@french\l@canadien
37      \fi
38    \else
39      \let\l@french\l@français
40      \fi
41 \fi

```

Now `\l@french` is always defined.

The internal name for the French language is `french`; `français` and `frenchb` are synonymous for `french`: first let both names use the same hyphenation patterns. Later we will have to set aliases for `\captionfrench`, `\datefrench`, `\extrasfrench` and `\noextrasfrench`. As French uses the standard values of `\lefthyphenmin` (2) and `\righthyphenmin` (3), no special setting is required here.

```

42 \ifx\l@français\@undefined
43   \let\l@français\l@french
44 \fi
45 \ifx\l@frenchb\@undefined
46   \let\l@frenchb\l@french
47 \fi

```

When this language definition file was loaded for one of the Canadian versions of French we need to make sure that a suitable hyphenation pattern register will be found by `TEX`.

```

48 \ifx\l@canadien\@undefined
49   \let\l@canadien\l@french
50 \fi
51 \ifx\l@acadian\@undefined
52   \let\l@acadian\l@french
53 \fi

```

This language definition can be loaded for different variants of the French language. The ‘key’ `babel` macros are only defined once, using ‘`french`’ as the language name, but `frenchb` and `français` are synonymous.

```

54 \def\datefrançais{\datefrench}
55 \def\datefrenchb{\datefrench}
56 \def\extrasfrançais{\extrasfrench}
57 \def\extrasfrenchb{\extrasfrench}
58 \def\noextrasfrançais{\noextrasfrench}
59 \def\noextrasfrenchb{\noextrasfrench}

```

`\extrasfrench` The macro `\extrasfrench` will perform all the extra definitions needed for the French language. The macro `\noextrasfrench` is used to cancel the actions of `\extrasfrench`.

In French, character “apostrophe” is a letter in expressions like `l’ambulance` (French hyphenation patterns provide entries for this kind of words). This means that the `\lccode` of “apostrophe” has to be non null in French for proper hyphenation of those expressions, and has to be reset to null when exiting French.

`\ifBUnicode` French hyphenation patterns are now coded in Unicode, see file `hyph-fr.tex`.  
`\ifLuaTeX` XeTeX and LuaTeX engines require some extra code to deal with the French “apos-  
`\ifXeTeX`

trophe”. Let’s define three new ‘if’: `\ifBLaTeX`, `\ifBTeX` and `\ifBUnicode` which will be true for XeTeX and LuaTeX engines and false for 8-bits engines.

```

60 \newif\ifBUnicode
61 \newif\ifBLaTeX
62 \newif\ifBTeX
63 \begingroup\expandafter\expandafter\expandafter\endgroup
64 \expandafter\ifx\csname luatexversion\endcsname\relax
65 \else
66   \BUnicodetrue \BLaTeXtrue
67 \fi
68 \begingroup\expandafter\expandafter\expandafter\endgroup
69 \expandafter\ifx\csname XeTeXrevision\endcsname\relax
70 \else
71   \BUnicodetrue \BTeXtrue
72 \fi

```

These `\lccode` changes will ensure correct hyphenation of words like *d’aventure*, *l’utopie*, with all TeX engines (XeTeX, LuaTeX, pdfTeX) using `hyph-fr.tex` patterns.

```

73 \@namedef{extras\CurrentOption}{\lccode'\='\'
74                               \ifBLaTeX \lccode'\="2019 \fi
75                               \ifBTeX \lccode"2019='\' \fi}
76 \@namedef{noextras\CurrentOption}{\lccode'\'=0
77                               \ifBTeX \lccode"2019=0 \fi}

```

One more thing `\extrasfrench` needs to do is to make sure that “Frenchspacing” is in effect. `\noextrasfrench` will switch “Frenchspacing” off again if necessary.

```

78 \addto\extrasfrench{\bbl@frenchspacing}
79 \addto\noextrasfrench{\bbl@nonfrenchspacing}

```

## 1.7 Punctuation

As long as no better solution is available, the ‘double punctuation’ characters (; ! ? and :) have to be made `\active` for an automatic control of the amount of space to insert before them. XeTeX provides an alternative to active characters and LuaTeX will hopefully do so as well in the (near?) future.

Before doing so, we have to save the standard definition of `\makecaption` (which includes two ‘:’) to compare it later to its definition at the `\begin{document}`.

```

80 \long\def\STD@makecaption#1#2{%
81   \vskip\abovecaptionskip
82   \sbox\@tempboxa{#1: #2}%
83   \ifdim \wd\@tempboxa >\hsize
84     #1: #2\par
85   \else
86     \global \@minipagefalse
87     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
88   \fi
89   \vskip\belowcaptionskip}

```

According to the I.N. specifications, the ‘:’ requires a normal space before it, but some people prefer a `\thinspace` (just like the other three). We define `\Fcolonspace` to hold the required amount of space (user customisable). In case some users are not satisfied with `\thinspace`’s width, it is also customisable.

```

90 \newcommand*{\Fcolonspace}{\space}
91 \newcommand*{\Fthinspace}{\thinspace}

\ifF@active@punct Check the availability of \XeTeXinterchartokenstate and decide whether the
\ifF@xetex@punct ‘double punctuation’ characters (; ! ? and :) have to be made \active or not.

92 \newif\ifFB@active@punct \FB@active@puncttrue
93 \newif\ifFB@xetex@punct
94 \begingroup\expandafter\expandafter\expandafter\endgroup
95 \expandafter\ifx\csname XeTeXinterchartokenstate\endcsname\relax
96 \else
97 \FB@xetex@puncttrue\FB@active@punctfalse
98 \fi

```

If `\XeTeXinterchartokenstate` is available, we use the “inter char” mechanism (as in `polyglossia`, see `gloss-french.ldf`) to provide correct spacing in French before the four characters ; ! ? and :. We use the same mechanism for French quotes (« and »), when automatic spacing for quotes is required by options `og=` and `fg=` in `\frenchbsetup{}` (see section 1.18).

For every character used in French text-mode (except spaces), `\XeTeXcharclass` value must be 0. `\XeTeXcharclass` value for spaces is assumed to be 255. Otherwise, the spacing before the ‘double punctuation’ characters and inside quotes might not be correct.

We switch `\XeTeXinterchartokenstate` to 1 and change the `\XeTeXcharclass` values of ; ! ? : ( ] « and » when entering French. Special care is taken to restore them to their initial values when leaving French.

```

99 \newif\ifFBAutoSpaceGuill \FBAutoSpaceGuilltrue
100 \newif\ifFBguillo@addspace
101 \newif\ifFBguillf@addspace
102 \newif\ifFBog@addspace \FBog@addspace true
103 \newif\ifFBfg@addspace \FBfg@addspace true
104 \newcount\FB@interchartokenstateORI
105 \ifFB@xetex@punct
106 \ifLaTeXe
107 \PackageInfo{frenchb.ldf}{No need for active punctuation characters
108 \MessageBreak with this version of XeTeX! reported}
109 \fi

```

We will need the following code (borrowed from `zhsusefulmacros.sty`) for loops:

```

110 \ifundefined{@for}{%
111 \def\@nnil{\@nil}%
112 \def\@empty{}%
113 \def\@fornoop#1\@#2#3{%
114 \long\def\@for#1:=#2\do#3{%
115 \expandafter\def\expandafter\@fortmp\expandafter{#2}%
116 \ifx\@fortmp\@empty \else
117 \expandafter\@forloop#2,\@nil,\@nil\@#1{#3}\fi}%
118 \long\def\@forloop#1,#2,#3\@#4#5{\def#4{#1}\ifx #4\@nnil \else
119 #5\def#4{#2}\ifx #4\@nnil \else#5\@iforloop #3\@#4{#5}\fi\fi}%
120 \long\def\@iforloop#1,#2\@#3#4{\def#3{#1}\ifx #3\@nnil
121 \expandafter\@fornoop \else
122 #4\relax\expandafter\@iforloop\fi#2\@#3{#4}}%
123 \def\@tfor#1:={\@tfor#1 }%
124 \long\def\@tfor#1#2\do#3{\def\@fortmp{#2}\ifx\@fortmp\space\else

```

```

125 \@tforloop#2\@nil\@nil\@#1{#3}\fi}%
126 \long\def\@tforloop#1#2\@#3#4{\def#3{#1}\ifx #3\@nnil
127 \expandafter\@fornoop \else
128 #4\relax\expandafter\@tforloop\fi#2\@#3{#4}}%
129 }-}%

```

Five new character classes are defined for `frenchb`.

```

130 \newXeTeXintercharclass\FB@punctthick
131 \newXeTeXintercharclass\FB@punctthin
132 \newXeTeXintercharclass\FB@punctnul
133 \newXeTeXintercharclass\FB@punctguilo
134 \newXeTeXintercharclass\FB@punctguilf

```

We define a command to store the `\XeTeXcharclass` values which will be modified for French (as a comma separated list) and a command to retrieve them.

```

135 \def\FB@charclassesORI{}
136 \def\empty{}
137 \def\FB@parse#1,#2\endparse{\def\FB@class{#1}%
138 \def\FB@charclassesORI{#2}}%

```

`\FB@xetex@punct@french` The following command will be executed when entering French, it first saves the values to be modified, then fits them to our needs. It also redefines `\shorthandoff` and `\shorthandon` (locally) to avoid error messages with XeTeX-based engines.

```
139 \newcommand*{\FB@xetex@punct@french}{%
```

Saving must not be repeated if saved values are already in.

```

140 \ifx\FB@charclassesORI\empty
141 \FB@interchartokenstateORI=\XeTeXinterchartokenstate
142 \@for\FB@char:={'\:', '\;', '\!', '\?', "AB", "BB", "\(\, \'[, '{, '\,, '\., %
143             '\-, '\), '\], '\}', '\%', "22,"27,"60,"2019}\do
144         {\edef\FB@charclassesORI{\FB@charclassesORI%
145                                     \the\xeTeXcharclass\FB@char,}}%
146     \let\shorthandonORI\shorthandon
147     \let\shorthandoFFORI\shorthandoFF
148 \fi
```

Set the classes and interactions between classes.

```

149 \XeTeXinterchartokenstate=1
150 \XeTeXcharclass '\: = \FB@punctthick
151 \XeTeXinterchartoks \z@ \FB@punctthick = {%
152     \ifhmode\FDP@colonspace\fi}%
153 \XeTeXinterchartoks \FB@punctguilf \FB@punctthick = {%
154     \FDP@colonspace}%
155 \XeTeXinterchartoks 255 \FB@punctthick = {%
156     \ifhmode\unskip\penalty\@M\Fcolonspace\fi}%
157 \@for\FB@char:={'\;,'!\,'?}\do
158     {\XeTeXcharclass\FB@char=\FB@punctthin}%
159 \XeTeXinterchartoks \z@ \FB@punctthin = {%
160     \ifhmode\FDP@thinspace\fi}%
161 \XeTeXinterchartoks \FB@punctguilf \FB@punctthin = {%
162     \FDP@thinspace}%
163 \XeTeXinterchartoks 255 \FB@punctthin = {%
164     \ifhmode\unskip\penalty\@M\Fthinspace\fi}%
165 \XeTeXinterchartoks \FB@punctguilo \z@ = {%
166     \ifFBAutoSpaceGuill\FBguill@spacing\fi}%

```

```

167 \XeTeXinterchartoks \FB@punctguilo 255 = {%
168     \ifFBAutoSpaceGuill\FBguill@spacing\ignorespaces\fi}%
169 \XeTeXinterchartoks \z@ \FB@punctguilf = {%
170     \ifFBAutoSpaceGuill\FBguill@spacing\fi}%
171 \XeTeXinterchartoks \FB@punctthin \FB@punctguilf = {%
172     \ifFBAutoSpaceGuill\FBguill@spacing\fi}%
173 \XeTeXinterchartoks 255 \FB@punctguilf = {%
174     \ifFBAutoSpaceGuill\unskip\FBguill@spacing\fi}%

```

This avoids spurious spaces in (!), [?], etc.

```

175 \@for\FB@char:={'\[, '\{, '\{ } \do
176     {\XeTeXcharclass\FB@char=\FB@punctnul}%

```

These characters have their class changed by `xeCJK.sty`, let's reset them to 0 in French.

```

177 \@for\FB@char:={'\{, '\., '\., '\., '\., '\., '\., '\., '\., '\., \%
178     "22,"27,"60,"2019\do
179     {\XeTeXcharclass\FB@char=\z@}%

```

With Xe(La)TeX, French defines no active shorthands.

```

180 \def\shorthandoff##1{%
181     \ifundefined{PackageWarning}%
182     {\let\PackageWarning\undefined}%
183     {\PackageWarning{frenchb.ldf}{\protect\shorthandoff{;!?} is
184     helpless with XeTeX,\MessageBreak use \protect\NoAutoSpacing
185     \space *inside a group* instead;\MessageBreak reported}%
186     }%
187     }%
188 \def\shorthandon##1{%
189     }

```

`\FB@xetex@punct@nonfrench` The following command will be executed when leaving French for restoring classes and commands modified in French. When French is not the main language, `\noextrasfrench` is executed 'AtBeginDocument', so the test on `\FB@charclassesORI` is mandatory.

```

190 \newcommand*{\FB@xetex@punct@nonfrench}{%
191     \ifx\FB@charclassesORI\empty
192     \else
193     \@for\FB@char:={'\:, '\;, '\!, '\?, "AB,"BB, '\(, '\[, '\{, '\., '\., \%
194     '\., '\., '\., '\., "22,"25,"27,"60,"2019\do
195     {\expandafter\FB@parse\FB@charclassesORI\endparse
196     \XeTeXcharclass\FB@char=\FB@class}%
197     \def\FB@charclassesORI{%
198     \XeTeXinterchartokenstate=\FB@interchartokenstateORI
199     \let\shorthandon\shorthandonORI
200     \let\shorthandoff\shorthandoffORI
201     \fi
202     }
203     \addto\extrasfrench{\FB@xetex@punct@french}
204     \addto\noextrasfrench{\FB@xetex@punct@nonfrench}
205 \fi

```

Otherwise we need to make the four characters ; ! ? and : 'active' and provide their definitions.

```

206 \ifFB@active@punct

```

```

207 \initiate@active@char{:}%
208 \initiate@active@char{;}%
209 \initiate@active@char{!}%
210 \initiate@active@char{?}%

```

We first tune the amount of space before ; ! ? and :. This should only happen in horizontal mode, hence the test `\ifhmode`.

In horizontal mode, if a space has been typed before ‘;’ we remove it and put an unbreakable `\thinspace` instead. If no space has been typed, we add `\FDP@thinspace` which will be defined, up to the user’s wishes, as an automatic added thin space, or as `\@empty`.

```

211 \declare@shorthand{french}{;}{%
212     \ifhmode
213     \ifdim\lastskip>\z@
214         \unskip\penalty\@M\Fthinspace
215     \else
216         \FDP@thinspace
217     \fi
218 \fi

```

Now we can insert a ; character.

```

219 \string;}

```

The next three definitions are very similar.

```

220 \declare@shorthand{french}{!}{%
221     \ifhmode
222     \ifdim\lastskip>\z@
223         \unskip\penalty\@M\Fthinspace
224     \else
225         \FDP@thinspace
226     \fi
227 \fi
228 \string!}
229 \declare@shorthand{french}{?}{%
230     \ifhmode
231     \ifdim\lastskip>\z@
232         \unskip\penalty\@M\Fthinspace
233     \else
234         \FDP@thinspace
235     \fi
236 \fi
237 \string?}
238 \declare@shorthand{french}{:}{%
239     \ifhmode
240     \ifdim\lastskip>\z@
241         \unskip\penalty\@M\Fcolonspace
242     \else
243         \FDP@colonspace
244     \fi
245 \fi
246 \string:}

```

When the active characters appear in an environment where their French behaviour is not wanted they should give an ‘expected’ result. Therefore we define shorthands at system level as well.

```

247 \declare@shorthand{system}{:}{\string:}
248 \declare@shorthand{system}{!}{\string!}
249 \declare@shorthand{system}{?}{\string?}
250 \declare@shorthand{system}{;}{\string;}
251 %}

```

We specify that the French group of shorthands should be used when switching to French.

```

252 \addto\extrasfrench{%
253   \languageshorthands{french}%

```

These characters are ‘turned on’ once, later their definition may vary. Don’t misunderstand the following code: they keep being active all along the document, even when leaving French.

```

254   \bbl@activate{:}\bbl@activate{;}%
255   \bbl@activate{!}\bbl@activate{?}%
256 }
257 \addto\noextrasfrench{%
258   \bbl@deactivate{:}\bbl@deactivate{;}%
259   \bbl@deactivate{!}\bbl@deactivate{?}}
260 \fi

```

A new ‘if’ `\FBAutoSpacePunctuation` needs to be defined now.

```

261 \newif\ifFBAutoSpacePunctuation \FBAutoSpacePunctuationtrue

```

`\AutoSpaceBeforeFDP` and `\FDP@thinspace` and `\FDP@colonspace` are defined as unbreakable spaces by `\autospace@beforeFDP` or as `\@empty` by `\noautospace@beforeFDP` (internal commands), user commands `\AutoSpaceBeforeFDP` and `\NoAutoSpaceBeforeFDP` do the same and take care of the flag `\ifFBAutoSpacePunctuation` in  $\text{\LaTeX}$ . Set the default now for Plain (done later for  $\text{\LaTeX}$ ).

```

262 \def\autospace@beforeFDP{%
263   \def\FDP@thinspace{\penalty\@M\Fthinspace}%
264   \def\FDP@colonspace{\penalty\@M\Fcolonspace}%
265 \def\noautospace@beforeFDP{\let\FDP@thinspace\@empty
266   \let\FDP@colonspace\@empty}
267 \ifLaTeXe
268   \def\AutoSpaceBeforeFDP{\autospace@beforeFDP
269     \FBAutoSpacePunctuationtrue}
270   \def\NoAutoSpaceBeforeFDP{\noautospace@beforeFDP
271     \FBAutoSpacePunctuationfalse}
272 \else
273   \let\AutoSpaceBeforeFDP\autospace@beforeFDP
274   \let\NoAutoSpaceBeforeFDP\noautospace@beforeFDP
275 \fi
276 \AutoSpaceBeforeFDP

```

In  $\text{\LaTeX 2}_{\epsilon}$  `\ttfamily` (and hence `\texttt`) will be redefined ‘AtBeginDocument’ as `\ttfamilyFB` so that no space is added before the four `;` `:` `!` `?` characters, even if `\AutoSpacePunctuation` is true. `\rmfamily` and `\sffamily` need to be redefined also (`\ttfamily` is not always used inside a group, its effect can be cancelled by `\rmfamily` or `\sffamily`).

These redefinitions can be canceled if necessary, for instance to recompile older documents, see option `OriginalTypewriter` below.



To be consistent with what is done for the ; : ! ? characters, `\ttfamilyFB` also switches off insertion of spaces inside French guillemets *when they are typed in as characters* with the ‘og’/‘fg’ options in `\frenchbsetup{}`. This is also a workaround for the weird behaviour of these characters in verbatim mode.

```

277 \ifLaTeXe
278   \let\ttfamilyORI\ttfamily
279   \let\rmfamilyORI\rmfamily
280   \let\sffamilyORI\sffamily
281   \DeclareRobustCommand\ttfamilyFB{%
282     \FBAutoSpaceGuillfalse
283     \FBog@addspacetrue      \FBfg@addspacetrue
284     \noautospace@beforeFDP\ttfamilyORI}%
285   \DeclareRobustCommand\rmfamilyFB{%
286     \FBAutoSpaceGuilltrue
287     \ifFBguillo@addspace\FBog@addspacefalse\fi
288     \ifFBguillf@addspace\FBfg@addspacefalse\fi
289     \ifFBAutoSpacePunctuation
290       \autospace@beforeFDP
291     \else
292       \noautospace@beforeFDP
293     \fi
294     \rmfamilyORI}%
295   \DeclareRobustCommand\sffamilyFB{%
296     \FBAutoSpaceGuilltrue
297     \ifFBguillo@addspace\FBog@addspacefalse\fi
298     \ifFBguillf@addspace\FBfg@addspacefalse\fi
299     \ifFBAutoSpacePunctuation
300       \autospace@beforeFDP
301     \else
302       \noautospace@beforeFDP
303     \fi
304     \sffamilyORI}%
305 \fi

```

`\NoAutoSpacing` The following command will switch off active punctuation characters (if any) and disable automatic spacing for French quote characters. It is engine independent (works for  $\text{\TeX}$  and  $\text{\XeTeX}$  based engines) and is meant to be used inside a group.

```

306 \newcommand*{\NoAutoSpacing}{\FBAutoSpaceGuillfalse
307   \FBog@addspacetrue      \FBfg@addspacetrue
308   \ifFB@active@punct\shorthandoff{;:!?}\fi
309   \ifFB@xetex@punct\XeTeXinterchartokenstate=0\fi
310 }

```

## 1.8 Commands for French quotation marks

`\og` The top macros for quotation marks will be called `\og` (“ouvrez guillemets”) and `\fg` (“fermez guillemets”). Another option for typesetting quotes in multilingual texts is to use the package `csquotes.sty` and its command `\enquote`.

```

311 \newcommand*{\og}{\@empty}
312 \newcommand*{\fg}{\@empty}

```

`\guillemotleft`    L<sup>A</sup>T<sub>E</sub>X users are supposed to use 8-bit output encodings (T1, LY1,...) to typeset French, those who still stick to OT1 should call `aeguill.sty` or a similar package.

`\guillemotright`    In both cases the commands `\guillemotleft` and `\guillemotright` will print the French opening and closing quote characters from the output font. For XeLaTeX, `\guillemotleft` and `\guillemotright` are defined by package `xunicode.sty`. We will check ‘AtBeginDocument’ that the proper output encodings are in use (see end of section 1.18).

`\textquotedblleft`    We give the following definitions for non-LaTeX users only as a fall-back, they are welcome to change them for anything better.

`\textquotedblright`

```

313 \ifLaTeXe
314 \else
315   \ifFBunicode
316     \def\guillemotleft{{\char"00AB}}
317     \def\guillemotright{{\char"00BB}}
318     \def\textquotedblleft{{\char"201C}}
319     \def\textquotedblright{{\char"201D}}
320   \else
321     \def\guillemotleft{\leavevmode\raise0.25ex
322                           \hbox{$\scriptscriptstyle\ll$}}
323     \def\guillemotright{\raise0.25ex
324                           \hbox{$\scriptscriptstyle\gg$}}
325     \def\textquotedblleft{‘}
326     \def\textquotedblright{’}
327   \fi
328   \let\xspace\relax
329 \fi

```

The next step is to provide correct spacing after `\guillemotleft` and before `\guillemotright`: a space precedes and follows quotation marks but no line break is allowed neither *after* the opening one, nor *before* the closing one. `\FBguill@spacing` which does the spacing, has been fine tuned by Thierry Bouche. French quotes (including spacing) are printed by `\FB@og` and `\FB@fg`, the expansion of the top level commands `\og` and `\og` is different in and outside French. We’ll try to be smart to users of David Carlisle’s `xspace` package: if this package is loaded there will be no need for `{}` or `\` to get a space after `\fg`, otherwise `xspace` will be defined as `\relax` (done at the end of this file).

```

330 \newcommand*\FBguill@spacing{\penalty\@M\hskip.8\fontdimen2\font
331                               plus.3\fontdimen3\font
332                               minus.8\fontdimen4\font}
333 \DeclareRobustCommand*\FB@og{\leavevmode\guillemotleft
334                               \ifFBog@addspace\FBguill@spacing\fi}
335 \DeclareRobustCommand*\FB@fg{\ifdim\lastskip>\z@\unskip\fi
336                               \ifFBfg@addspace\FBguill@spacing\fi
337                               \guillemotright\xspace}

```

The top level definitions for French quotation marks are switched on and off through the `\extrasfrench` `\noextrasfrench` mechanism. Outside French, `\og` and `\fg` will typeset standard English opening and closing double quotes.

```

338 \ifLaTeXe
339   \def\bbl@frenchguillemets{\renewcommand*\og{\FB@og}%
340                               \renewcommand*\fg{\FB@fg}}
341   \def\bbl@nonfrenchguillemets{\renewcommand*\og{\textquotedblleft}%

```

```

342         \renewcommand*{\fg}{\ifdim\lastskip>z@\unskip\fi
343                     \textquotedblright}}
344 \else
345   \def\bbl@frenchguillemets{\let\og\FB@og
346                       \let\fg\FB@fg}
347   \def\bbl@nonfrenchguillemets{\def\og{\textquotedblleft}%
348                       \def\fg{\ifdim\lastskip>z@\unskip\fi\textquotedblright}}
349 \fi
350 \addto\extrasfrench{\bbl@frenchguillemets}
351 \addto\noextrasfrench{\bbl@nonfrenchguillemets}

```

## 1.9 Date in French

`\datefrench` The macro `\datefrench` redefines the command `\today` to produce French dates.

```

352 \@namedef{date\CurrentOption}{%
353   \def\today{\number\day}\ifnum1=\day {\ier}\fi \space
354   \ifcase\month
355     \or janvier\or f{\FB@eacute}vrier\or mars\or avril\or mai\or
356     juin\or juillet\or ao{\FB@circonflexe}t\or septembre\or
357     octobre\or novembre\or d{\FB@eacute}cembre\fi
358   \space \number\year}}

```

## 1.10 Extra utilities

Let's provide the French user with some extra utilities.

`\up` `\up` eases the typesetting of superscripts like ‘<sup>1<sup>er</sup></sup>’. Up to version 2.0 of `frenchb`  
`\fup` `\up` was just a shortcut for `\textsuperscript` in  $\text{\LaTeX} 2_{\epsilon}$ , but several users complained that `\textsuperscript` typesets superscripts too high and too big, so we now define `\fup` as an attempt to produce better looking superscripts. `\up` is defined as `\fup` but `\frenchbsetup{FrenchSuperscripts=false}` redefines `\up` as `\textsuperscript` for compatibility with previous versions.

When a font has built-in superscripts, the best thing to do is to just use them, otherwise `\fup` has to simulate superscripts by scaling and raising ordinary letters. Scaling is done using package `scalefnt` which will be loaded at the end of `babel`'s loading (`frenchb` being an option of `babel`, it cannot load a package while being read).

```

359 \newif\ifFB@poorman
360 \newdimen\FB@Mht
361 \ifLaTeXe
362   \AtEndOfPackage{\RequirePackage{scalefnt}}

```

`\FB@up@fake` holds the definition of fake superscripts. The scaling ratio is 0.65, raising is computed to put the top of lower case letters (like ‘m’) just under the top of upper case letters (like ‘M’), precisely 12% down. The chosen settings look correct for most fonts, but can be tuned by the end-user if necessary by changing `\FBsupR` and `\FBsupS` commands.

`\FB@lc` is defined as `\MakeLowercase` to inhibit the uppercasing of superscripts (this may happen in page headers with the standard classes but is wrong); `\FB@lc` can be redefined to do nothing by option `LowercaseSuperscripts=false` of `\frenchbsetup{}`.

```

363   \newcommand*{\FBsupR}{-0.12}

```

```

364 \newcommand*\FBsupS{0.65}
365 \newcommand*\FB@lc{1}{\MakeLowercase{#1}}
366 \DeclareRobustCommand*\FB@up@fake{1}{%
367   \settoheight{\FB@Mht}{M}%
368   \addtolength{\FB@Mht}{\FBsupR \FB@Mht}%
369   \addtolength{\FB@Mht}{-\FBsupS ex}%
370   \raisebox{\FB@Mht}{\scalefont{\FBsupS}{\FB@lc{#1}}}%
371 }

```

The only packages I currently know to take advantage of real superscripts are a) `xltxtra` used in conjunction with XeLaTeX and OpenType fonts having the font feature 'VerticalPosition=Superior' (`xltxtra` defines `\realsuperscript` and `\fakesuperscript`) and b) `fourier` (from version 1.6) when Expert Utopia fonts are available.

`\FB@up` checks whether the current font is a Type1 'Expert' (or 'Pro') font with real superscripts or not (the code works currently only with `fourier-1.6` but could work with any Expert Type1 font with built-in superscripts, see below), and decides to use real or fake superscripts. It works as follows: the content of `\f@family` (family name of the current font) is split by `\FB@split` into two pieces, the first three characters ('fut' for Fourier, 'ppl' for Adobe's Palatino, ...) stored in `\FB@firstthree` and the rest stored in `\FB@suffix` which is expected to be 'x' or 'j' for expert fonts.

```

372 \def\FB@split#1#2#3#4\@nil{\def\FB@firstthree{#1#2#3}%
373   \def\FB@suffix{#4}}
374 \def\FB@x{x}
375 \def\FB@j{j}
376 \DeclareRobustCommand*\FB@up{1}{%
377   \bgroup \FB@poormantrue
378   \expandafter\FB@split\f@family\@nil

```

Then `\FB@up` looks for a .fd file named `t1fut-sup.fd` (Fourier) or `t1ppl-sup.fd` (Palatino), etc. supposed to define the subfamily (`fut-sup` or `ppl-sup`, etc.) giving access to the built-in superscripts. If the .fd file is not found by `\IfFileExists`, `\FB@up` falls back on fake superscripts, otherwise `\FB@suffix` is checked to decide whether to use fake or real superscripts.

```

379   \edef\reserved@a{\lowercase{%
380     \noexpand\IfFileExists{\f@encoding\FB@firstthree -sup.fd}}}%
381   \reserved@a
382   {\ifx\FB@suffix\FB@x \FB@poormanfalse\fi
383     \ifx\FB@suffix\FB@j \FB@poormanfalse\fi
384     \if\FB@poorman \FB@up@fake{#1}%
385     \else \FB@up@real{#1}%
386     \fi}%
387   {\FB@up@fake{#1}}%
388   \egroup}

```

`\FB@up@real` just picks up the superscripts from the subfamily (and forces lowercase).

```

389 \newcommand*\FB@up@real{1}{\bgroup
390   \fontfamily{\FB@firstthree -sup}\selectfont \FB@lc{#1}\egroup}

```

`\fup` is now defined as `\FB@up` unless `\realsuperscript` is defined (occurs with XeLaTeX calling `xltxtra.sty`).

```

391 \DeclareRobustCommand*\fup{1}{%

```

```

392 \ifundefined{realsuperscript}%
393   {\FB@up{#1}\let\realsuperscript\undefined}%
394   {\bgroup\let\fakesuperscript\FB@up@fake
395    \realsuperscript{\FB@lc{#1}}\egroup}}

```

Let's provide a temporary definition for `\up` (redefined 'AtBeginDocument' as `\fup` or `\textsuperscript` according to `\frenchbsetup{}` options).

```

396 \providecommand*\up{\relax}

```

Poor man's definition of `\up` for Plain.

```

397 \else
398 \providecommand*\up[1]{\leavevmode\raise1ex\hbox{\sevenrm #1}}
399 \fi

```

`\ieme` Some handy macros for those who don't know how to abbreviate ordinals:

```

\ier 400 \def\ieme{\up{\lowercase{e}}\xspace}
\iere 401 \def\ienes{\up{\lowercase{es}}\xspace}
\iemes 402 \def\ier{\up{\lowercase{er}}\xspace}
\iers 403 \def\iers{\up{\lowercase{ers}}\xspace}
\ieres 404 \def\iere{\up{\lowercase{re}}\xspace}
405 \def\ieres{\up{\lowercase{res}}\xspace}

```

`\No` And some more macros relying on `\up` for numbering, first two support macros.

```

\no 406 \newcommand*\FrenchEnumerate[1]{%
\Nos 407     #1\up{\lowercase{o}}\kern+.3em}
\nos 408 \newcommand*\FrenchPopularEnumerate[1]{%
\primo 409     #1\up{\lowercase{o}})\kern+.3em}

```

`\fprimo`) Typing `\primo` should result in '1°',

```

410 \def\primo{\FrenchEnumerate1}
411 \def\secundo{\FrenchEnumerate2}
412 \def\tertio{\FrenchEnumerate3}
413 \def\quarto{\FrenchEnumerate4}

```

while typing `\fprimo` gives '1°'.

```

414 \def\fprimo{\FrenchPopularEnumerate1}
415 \def\fsecundo{\FrenchPopularEnumerate2}
416 \def\ftertio{\FrenchPopularEnumerate3}
417 \def\fquarto{\FrenchPopularEnumerate4}

```

Let's provide four macros for the common abbreviations of "Numéro".

```

418 \DeclareRobustCommand*\No{\N\up{\lowercase{o}}\kern+.2em}
419 \DeclareRobustCommand*\no{\n\up{\lowercase{o}}\kern+.2em}
420 \DeclareRobustCommand*\Nos{\N\up{\lowercase{os}}\kern+.2em}
421 \DeclareRobustCommand*\nos{\n\up{\lowercase{os}}\kern+.2em}

```

`\bsc` As family names should be written in small capitals and never be hyphenated, we provide a command (its name comes from Boxed Small Caps) to input them easily. Note that this command has changed with version 2 of `frenchb`: a `\kern0pt` is used instead of `\hbox` because `\hbox` would break microtype's font expansion; as a (positive?) side effect, composed names (such as Dupont-Durand) can now be hyphenated on explicit hyphens. Usage: `Jean~\bsc{Duchemin}`.

```

422 \DeclareRobustCommand*\bsc[1]{\leavevmode\begingroup\kern0pt
423     \scshape #1\endgroup}
424 \ifLaTeXe\else\let\scshape\relax\fi

```

Some definitions for special characters. We won't define `\tilde` as a Text Symbol not to conflict with the macro `\tilde` for math mode and use the name `\tild` instead. Note that `\boi` may *not* be used in math mode, its name in math mode is `\backslash`. `\degree` can be accessed by the command `\r{}` for ring accent.

```

425 \ifFBunicode
426   \newcommand*{\at}{\char"0040}}
427   \newcommand*{\circonflexe}{\char"005E}}
428   \newcommand*{\tild}{\char"007E}}
429   \newcommand*{\boi}{\textbackslash}
430   \newcommand*{\degree}{\char"00B0}}
431 \else
432   \ifLaTeXe
433     \DeclareTextSymbol{\at}{T1}{64}
434     \DeclareTextSymbol{\circonflexe}{T1}{94}
435     \DeclareTextSymbol{\tild}{T1}{126}
436     \DeclareTextSymbolDefault{\at}{T1}
437     \DeclareTextSymbolDefault{\circonflexe}{T1}
438     \DeclareTextSymbolDefault{\tild}{T1}
439     \DeclareRobustCommand*{\boi}{\textbackslash}
440     \DeclareRobustCommand*{\degree}{\r{}}
441   \else
442     \def\T@one{T1}
443     \ifx\f@encoding\T@one
444       \newcommand*{\degree}{\char6}}
445     \else
446       \newcommand*{\degree}{\char23}}
447     \fi
448     \newcommand*{\at}{\char64}}
449     \newcommand*{\circonflexe}{\char94}}
450     \newcommand*{\tild}{\char126}}
451     \newcommand*{\boi}{\backslash}
452   \fi
453 \fi

```

French dates and captions make use of four non-ascii characters (à, è, é and û). This is fine except for (plain) XeTeX (`\accent` commands are not implemented), so we define four new commands to deal with this issue.

```

454 \newcommand*{\FBgrave}{\'a}
455 \newcommand*{\FBgrave}{\'e}
456 \newcommand*{\FBacute}{\'e}
457 \newcommand*{\FBucirconflexe}{\^u}
458 \ifFBunicode
459   \ifLaTeXe
460   \else
461     \def\FBgrave{\char"00E0}}
462     \def\FBgrave{\char"00E8}}
463     \def\FBacute{\char"00E9}}
464     \def\FBucirconflexe{\char"00FB}}
465   \fi
466 \fi

```

`\degrees` We now define a macro `\degrees` for typesetting the abbreviation for ‘degrees’ (as in ‘degrees Celsius’). As the bounding box of the character ‘degree’ has *very* different widths in CM/EC and PostScript fonts, we fix the width of the bounding

box of `\degrees` to 0.3em, this lets the symbol ‘degree’ stick to the preceding (e.g., `45\degrees`) or following character (e.g., `20~\degrees C`).

If T<sub>E</sub>X Companion fonts are available (`textcomp.sty`), we pick up `\textdegree` from them instead of using emulating ‘degrees’ from the `\r{}` accent. Otherwise we overwrite the (poor) definition of `\textdegree` given in `latin1.def`, `applemac.def` etc. (called by `inputenc.sty`) by our definition of `\degrees`. We also advice the user (once only) to use TS1-encoding.

```

467 \ifLaTeXe
468   \newcommand*{\degrees}{\degree}
469   \ifBUnicode
470     \DeclareRobustCommand*{\degrees}{\degree}
471   \else
472     \def\Warning@degree@TSone{%
473       \PackageWarning{frenchb.ldf}{%
474         Degrees would look better in TS1-encoding:
475         \MessageBreak add \protect
476         \usepackage{textcomp} to the preamble.
477         \MessageBreak Degrees used}}
478     \AtBeginDocument{\@ifundefined{DeclareEncodingSubset}%
479       {\DeclareRobustCommand*{\degrees}{%
480         \leavevmode\hbox to 0.3em{\hss\degree\hss}%
481         \Warning@degree@TSone
482         \global\let\Warning@degree@TSone\relax}%
483       \let\textdegree\degrees}%
484       {\DeclareRobustCommand*{\degrees}{%
485         \hbox{\UseTextSymbol{TS1}{\textdegree}}}}}%
486     }
487   \fi
488 \else
489   \newcommand*{\degrees}{%
490     \leavevmode\hbox to 0.3em{\hss\degree\hss}}
491 \fi

```

## 1.11 Formatting numbers

`\DecimalMathComma` As mentioned in the T<sub>E</sub>Xbook p. 134, the comma is of type `\mathpunct` in math mode: it is automatically followed by a space. This is convenient in lists and intervals but unpleasant when the comma is used as a decimal separator in French: it has to be entered as `{,}`. `\DecimalMathComma` makes the comma be an ordinary character (of type `\mathord`) in French *only* (no space added); `\StandardMathComma` switches back to the standard behaviour of the comma.

```

492 \newcount\std@mcc
493 \newcount\dec@mcc
494 \std@mcc=\mathcode'\,
495 \dec@mcc=\std@mcc
496 \@tempcnta=\std@mcc
497 \divide\@tempcnta by "1000
498 \multiply\@tempcnta by "1000
499 \advance\dec@mcc by -\@tempcnta
500 \newcommand*{\DecimalMathComma}{\iflanguage{french}%
501   {\mathcode'\,=\dec@mcc}{}}%
502   \addto\extrasfrench{\mathcode'\,=\dec@mcc}}

```

```

503 \newcommand*{\StandardMathComma}{\mathcode'\,=\std@mcc
504   \addto\extrasfrench{\mathcode'\,=\std@mcc}}
505 \addto\noextrasfrench{\mathcode'\,=\std@mcc}

```

**\nombre** The command `\nombre` is now borrowed from `numprint.sty` for  $\text{\LaTeX} 2_{\epsilon}$ . There is no point to maintain the former tricky code when a package is dedicated to do the same job and more. For Plain based formats, `\nombre` no longer formats numbers, it prints them as is and issues a warning about the change. Fake command `\nombre` for Plain based formats, warning users of `frenchb` v. 1.x. of the change.

```

506 \newcommand*{\nombre}[1]{\#1}\message{%
507   *** \noexpand\nombre no longer formats numbers\string! ***}}%

```

The next definitions only make sense for  $\text{\LaTeX} 2_{\epsilon}$ . Let's cleanup and exit if the format is Plain based.

```

508 \let\FBstop@here\relax
509 \def\FBclean@on@exit{\let\ifLaTeXe\undefined
510   \let\LaTeXettrue\undefined
511   \let\LaTeXefalse\undefined}
512 \ifx\magnification\@undefined
513 \else
514   \def\FBstop@here{\let\STD@makecaption\relax
515     \FBclean@on@exit
516     \ldf@quit\CurrentOption\endinput}
517 \fi
518 \FBstop@here

```

What follows now is for  $\text{\LaTeX} 2_{\epsilon}$  *only*. We redefine `\nombre` for  $\text{\LaTeX} 2_{\epsilon}$ . A warning is issued at the first call of `\nombre` if `\numprint` is not defined, suggesting what to do. The package `numprint` is *not* loaded automatically by `frenchb` because of possible options conflict.

```

519 \renewcommand*{\nombre}[1]{\Warning@nombre\numprint{#1}}
520 \newcommand*{\Warning@nombre}{%
521   \@ifundefined{numprint}%
522     {\PackageWarning{frenchb.ldf}{%
523       \protect\nombre\space now relies on package numprint.sty,
524       \MessageBreak add \protect
525       \usepackage[autolanguage]{numprint}\MessageBreak
526       to your preamble *after* loading babel, \MessageBreak
527       see file numprint.pdf for more options.\MessageBreak
528       \protect\nombre\space called}%
529     \global\let\Warning@nombre\relax
530     \global\let\numprint\undefined
531   }{}%
532 }

533 \newcommand*{\ThinSpaceInFrenchNumbers}{%
534   \PackageWarning{frenchb.ldf}{%
535     Type \protect\frenchbsetup{ThinSpaceInFrenchNumbers}
536     \MessageBreak Command \protect\ThinSpaceInFrenchNumbers\space
537     is no longer\MessageBreak defined in frenchb v.~2,}}

```



## 1.12 Caption names

The next step consists of defining the French equivalents for the L<sup>A</sup>T<sub>E</sub>X caption names.

`\captionsfrench` Let's first define `\captionsfrench` which sets all strings used in the four standard document classes provided with L<sup>A</sup>T<sub>E</sub>X.

```

538 \@namedef{captions\CurrentOption}{%
539   \def\refname{R{\FBacutef}{\FBacutereferences}%
540   \def\abstractname{R{\FBacutef}{\FBacutefsum{\FBacutef}}%
541   \def\bibname{Bibliographie}%
542   \def\prefacename{Pr{\FBacutef}{\FBacuteface}%
543   \def\chaptername{Chapitre}%
544   \def\appendixname{Annexe}%
545   \def\contentsname{Table des mati{\FBegraveres}%
546   \def\listfigurename{Table des figures}%
547   \def\listtablename{Liste des tableaux}%
548   \def\indexname{Index}%
549   \def\figurename{{\scshape Figure}}%
550   \def\tablename{{\scshape Table}}%

  "Première partie" instead of "Part I".

551   \def\partname{\protect\@Fpt partie}%
552   \def\@Fpt{{\ifcase\value{part}\or Premi{\FBegraverere\or
553     Deuxi{\FBegraveme\or Troisi{\FBegraveme\or
554     Quatri{\FBegraveme\or Cinqui{\FBegraveme\or
555     Sixi{\FBegraveme\or Septi{\FBegraveme\or Huiti{\FBegraveme\or
556     Neuvi{\FBegraveme\or Dixi{\FBegraveme\or Onzi{\FBegraveme\or
557     Douzi{\FBegraveme\or Treizi{\FBegraveme\or
558     Quatorzi{\FBegraveme\or Quinzi{\FBegraveme\or
559     Seizi{\FBegraveme\or Dix-septi{\FBegraveme\or
560     Dix-huiti{\FBegraveme\or Dix-neuvi{\FBegraveme\or
561     Vingt{\FBegraveme\fi}\space\def\thepart{}}%
562   \def\pagename{page}%
563   \def\seename{voir}%
564   \def\alsoname{voir aussi}%
565   \def\enclname{P.~J. }%
566   \def\ccname{Copie {\FBagrave} }%
567   \def\headtoname{}%
568   \def\proofname{D{\FBacutef}{\FBacutemonstration}%
569   \def\glossaryname{Glossaire}%
570 }
```

As some users who choose `frenchb` or `francais` as option of `babel`, might customise `\captionsfrenchb` or `\captionsfrancais` in the preamble, we merge their changes at the `\begin{document}` when they do so. The other variants of French (canadien, acadian) are defined by checking if the relevant option was used and then adding one extra level of expansion.

```

571 \@AtBeginDocument{\let\captions@French\captionsfrench
572   \ifundefined{captionsfrenchb}%
573     {\let\captions@Frenchb\relax}%
574     {\let\captions@Frenchb\captionsfrenchb}%
575   \ifundefined{captionsfrancais}%
576     {\let\captions@Francais\relax}%

```

```

577         {\let\captions@Francais\captionsfrancais}%
578         \def\captionsfrench{\captions@French
579         \captions@Francais\captions@Frenchb}%
580         \def\captionsfrancais{\captionsfrench}%
581         \def\captionsfrenchb{\captionsfrench}%
582     }
583 \ifpackagewith{babel}{canadien}{%
584     \def\captionscanadien{\captionsfrench}%
585     \def\datecanadien{\datefrench}%
586     \def\extrascanadien{\extrasfrench}%
587     \def\noextrascanadien{\noextrasfrench}%
588 }{}
589 \ifpackagewith{babel}{acadian}{%
590     \def\captionsacadian{\captionsfrench}%
591     \def\dateacadian{\datefrench}%
592     \def\extrasacadian{\extrasfrench}%
593     \def\noextrasacadian{\noextrasfrench}%
594 }{}

```

`\CaptionSeparator` Let's consider now captions in figures and tables. In French, captions in figures and tables should be printed with endash (‘–’) instead of the standard ‘:’. The standard definition of `\@makecaption` (e.g., the one provided in `article.cls`, `report.cls`, `book.cls` which is frozen for L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> according to Frank Mittelbach), has been saved in `\STD@makecaption` before making ‘:’ active (see section 1.7). ‘AtBeginDocument’ we compare it to its current definition (some classes like `koma-script` classes, `AMS` classes, `ua-thesis.cls`... change it). If they are identical, `frenchb` just adds a hook called `\CaptionSeparator` to `\@makecaption`, `\CaptionSeparator` defaults to ‘: ’ as in the standard `\@makecaption`, and will be changed to ‘– ’ in French. If the definitions differ, `frenchb` doesn't overwrite the changes, but prints a message in the `.log` file.

```

595 \newcommand{\FBWarning}[2]{\PackageWarning{#1}{#2}}
596 \def\CaptionSeparator{\string:\space}
597 \long\def\FB@makecaption#1#2{%
598     \vskip\abovecaptionskip
599     \sbox\@tempboxa{#1\CaptionSeparator #2}%
600     \ifdim \wd\@tempboxa >\hsize
601         #1\CaptionSeparator #2\par
602     \else
603         \global \@minipagefalse
604         \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
605     \fi
606     \vskip\belowcaptionskip}
607 \AtBeginDocument{%
608     \ifx\@makecaption\STD@makecaption
609         \global\let\@makecaption\FB@makecaption
610     \else
611         \ifundefined{@makecaption}%
612             {\let\@makecaption\undefined}%
613             {\FBWarning{frenchb.ldf}%
614             {The definition of \protect\@makecaption\space
615             has been changed,\MessageBreak
616             frenchb will NOT customise it;\MessageBreak reported}%
617         }%

```

```

618 \fi
619 \let\FB@makecaption\relax
620 \let\STD@makecaption\relax
621 }
622 \addto\extrasfrench{%
623   \def\CaptionSeparator{\space\textendash\space}}
624 \addto\noextrasfrench{%
625   \def\CaptionSeparator{\string:\space}}

```

### 1.13 French lists

`\listFB` Vertical spacing in lists should be shorter in French texts than the defaults provided by L<sup>A</sup>T<sub>E</sub>X. Note that the easy way, just changing values of vertical spacing parameters when entering French and restoring them to their defaults on exit would not work; so we define the command `\FB@listVsettings` to hold the settings to be used by the French variant `\listFB` of `\list`. Note that switching to `\listFB` reduces vertical spacing in *all* environments built on `\list`: `itemize`, `enumerate`, `description`, but also `abstract`, `quotation`, `quote` and `verse`...

The amount of vertical space before and after a list is given by `\topsep` + `\parskip` (+ `\partopsep` if the list starts a new paragraph). IMHO, `\parskip` should be added *only* when the list starts a new paragraph, so I subtract `\parskip` from `\topsep` and add it back to `\partopsep`; this will normally make no difference because `\parskip`'s default value is 0pt, but will be noticeable when `\parskip` is *not* null.

`\endlist` is not redefined, but `\endlistORI` and `\endlistFB` will be defined as `\endlist` ‘AtBeginDocument’ for users who define their own lists using `\listORI` or `\listFB`.

```

626 \let\listORI\list
627 \def\FB@listVsettings{%
628   \setlength{\itemsep}{0.4ex plus 0.2ex minus 0.2ex}%
629   \setlength{\parsep}{0.4ex plus 0.2ex minus 0.2ex}%
630   \setlength{\topsep}{0.8ex plus 0.4ex minus 0.4ex}%
631   \setlength{\partopsep}{0.4ex plus 0.2ex minus 0.2ex}%

```

`\parskip` is of type ‘skip’, its mean value only (*not the glue*) should be subtracted from `\topsep` and added to `\partopsep`, so convert `\parskip` to a ‘dimen’ using `\@tempdima`.

```

632   \@tempdima=\parskip
633   \addtolength{\topsep}{-\@tempdima}%
634   \addtolength{\partopsep}{\@tempdima}%
635 }
636 \def\listFB#1#2{\listORI{#1}{\FB@listVsettings #2}}

```

Let's now consider French itemize-lists. They differ from those provided by the standard L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> classes:

- The ‘•’ is never used in French itemize-lists, an emdash ‘—’ or an en-dash ‘-’ is preferred for all levels. The item label to be used in French is stored in `\FrenchLabelItem`, it defaults to ‘—’ and can be changed using `\frenchbsetup{}` (see section 1.18).
- Vertical spacing between items, before and after the list, should be *null* with *no glue* added;

- In French the labels of itemize-lists are vertically aligned as follows:

Text starting at ‘parindent’  
 $\Leftarrow$  Leftmargin  
 – first item...  
   – first second level item  
   – next one...  
   – second item...

`\FrenchLabelItem` Default labels for French itemize-lists (same label for all levels):

```
\Frlabelitemi 637 \newcommand*\FrenchLabelItem{\textendash}
\Frlabelitemii 638 \newcommand*\Frlabelitemi{\FrenchLabelItem}
\Frlabelitemiii 639 \newcommand*\Frlabelitemii{\FrenchLabelItem}
\Frlabelitemiv 640 \newcommand*\Frlabelitemiii{\FrenchLabelItem}
641 \newcommand*\Frlabelitemiv{\FrenchLabelItem}
```

`\bbl@frenchlabelitems` `\bbl@frenchlabelitems` saves current itemize labels and changes them to their value in French. This code should never be executed twice in a row, so we need a new flag that will be set and reset by `\bbl@nonfrenchlabelitems` and `\bbl@frenchlabelitems`. `\bbl@frenchlabelitems` also sets reasonable default values for lengths `\labelwidthFB` and `\listindentFB` to be defined below, but these lengths can still be customised in the preamble.

```
642 \newif\ifFB@enterFrench \FB@enterFrenchtrue
643 \def\bbl@frenchlabelitems{%
644   \ifFB@enterFrench
645     \let\@ltiORI\labelitemi
646     \let\@ltiiORI\labelitemii
647     \let\@ltiiiORI\labelitemiii
648     \let\@ltivORI\labelitemiv
649     \let\labelitemi\Frlabelitemi
650     \let\labelitemii\Frlabelitemii
651     \let\labelitemiii\Frlabelitemiii
652     \let\labelitemiv\Frlabelitemiv
653     \ifdim\labelwidthFB<\z@
654       \settowidth{\labelwidthFB}{\FrenchLabelItem}%
655     \fi
656     \ifdim\listindentFB<\z@
657       \ifdim\parindent=\z@
658         \setlength{\listindentFB}{1.5em}%
659       \else
660         \setlength{\listindentFB}{\parindent}%
661       \fi
662     \fi
663     \FB@enterFrenchfalse
664   \fi
665 }
666 \def\bbl@nonfrenchlabelitems{%
667   \ifFB@enterFrench
668   \else
669     \let\Frlabelitemi\labelitemi
670     \let\Frlabelitemii\labelitemii
671     \let\Frlabelitemiii\labelitemiii
672     \let\Frlabelitemiv\labelitemiv
```

```

673     \let\labelitemi\@ltiORI
674     \let\labelitemii\@ltiiORI
675     \let\labelitemiii\@ltiiiORI
676     \let\labelitemiv\@ltivORI
677     \FB@enterFrenchtrue
678   \fi
679 }

```

`\listindentFB` `\labelwidthFB` Let's define two lengths `\listindentFB` and `\labelwidthFB` to customise lists' horizontal indentations. They are given silly values here (−1 pt) in order to eventually enable their customisation in the preamble. They will get reasonable defaults later when entering French (see `\bbl@frenchlabelitems`) unless they have been customised.

```

680 \newlength\listindentFB
681 \setlength{\listindentFB}{-1pt}
682 \newlength\labelwidthFB
683 \setlength{\labelwidthFB}{-1pt}

```

`\FB@listHsettings` `\FB@listHsettings` holds the new horizontal settings chosen for French lists itemize and enumerate starting with version 2.6a. They are based on the look requested in French for itemize-lists.

```

684 \newlength\leftmarginFB
685 \def\FB@listHsettings{%
686   \leftmarginFB\labelwidthFB
687   \advance\leftmarginFB \labelsep
688   \leftmargini\leftmarginFB
689   \advance\leftmargini \listindentFB
690   \leftmarginii\leftmarginFB
691   \leftmarginiii\leftmarginFB
692   \leftmarginiv\leftmarginFB
693   \leftmargin\cename leftmargin\romannumeral\the\@listdepth\endcsname
694 }

```

`\itemizeFB` New environment for French itemize-lists.

`\FB@itemizesettings` `\FB@itemizesettings` does two things: first suppress all vertical spaces including glue, then set horizontal indentations according to `\FB@listHsettings` unless option `ListOldLayout` is true (compatibility with lists up to v. 2.5k).

```

695 \def\FB@itemizesettings{%
696   \setlength{\itemsep}{\z@}%
697   \setlength{\parsep}{\z@}%
698   \setlength{\topsep}{\z@}%
699   \setlength{\partopsep}{\z@}%
700   \@tempdima=\parskip
701   \addtolength{\topsep}{-\@tempdima}%
702   \addtolength{\partopsep}{-\@tempdima}%
703   \ifFBListOldLayout
704     \settowidth{\labelwidth}{\cename\@itemitem\endcsname}%
705     \setlength{\leftmargin}{\labelwidth}%
706     \addtolength{\leftmargin}{\labelsep}%
707     \addtolength{\leftmargin}{\parindent}%
708   \else
709     \FB@listHsettings
710   \fi

```

711 }

The definition of `\itemizeFB` follows the one of `\itemize` in standard  $\text{\LaTeX} 2_{\epsilon}$  classes (see `ltlists.dtx`), spaces are customised by `\FB@itemizesettings`.  
`\enditemizeFB` will be defined as `\endlist=\endlistORI` ‘AtBeginDocument’.

```
712 \def\itemizeFB{%
713   \ifnum \@itemdepth >\thr@@\toodeep\else
714     \advance\@itemdepth\@ne
715     \edef\@itemitem{labelitem\romannumeral\the\@itemdepth}%
716     \expandafter
717     \listORI
718     \csname\@itemitem\endcsname
719     \FB@itemizesettings
720   \fi}
```

`\enumerateFB` The definition of `\enumerateFB`, new to version 2.6a, follows the one of `\enumerate` in standard  $\text{\LaTeX} 2_{\epsilon}$  classes (see `ltlists.dtx`), vertical spaces are customised (or not) via `\list` (`=\listFB` or `\listORI`) and horizontal spaces (leftmargins) are borrowed from `itemize` lists via `\FB@listHsettings`.  
`\endenumerateFB` will be defined as `\endlist=\endlistORI` ‘AtBeginDocument’.

```
721 \def\enumerateFB{%
722   \ifnum \@enumdepth >\thr@@\toodeep\else
723     \advance\@enumdepth\@ne
724     \edef\@enumctr{enum\romannumeral\the\@enumdepth}%
725     \expandafter
726     \list
727     \csname label\@enumctr\endcsname
728     {\FB@listHsettings
729      \usecounter\@enumctr\def\makelabel##1{\hss\llap{##1}}}%
730   \fi}
```

`\descriptionFB` Same tuning for the `description` environment (see the original definition in `classes.dtx`).

```
731 \def\descriptionFB{%
732   \list{}{\FB@listHsettings
733     \labelwidth\z@ \itemindent-\leftmargin
734     \let\makelabel\descriptionlabel}}
```

## 1.14 French indentation of sections

`\bbl@frenchindent` In French the first paragraph of each section should be indented, this is another  
`\bbl@nonfrenchindent` difference with US-English. This is controlled by the flag `\if@afterindent`.  
 We will need to save the value of the flag `\if@afterindent` ‘AtBeginDocument’ before eventually changing its value.

```
735 \def\bbl@frenchindent{\let\@afterindentfalse\@afterindenttrue
736   \@afterindenttrue}
737 \def\bbl@nonfrenchindent{\let\@afterindentfalse\@aifORI
738   \@afterindentfalse}
```

## 1.15 Formatting footnotes

The `bigfoot` package deeply changes the way footnotes are handled. When `bigfoot` is loaded, we just warn the user that `frenchb` will drop the customisation of footnotes.

The layout of footnotes is controlled by two flags `\ifFBAutoSpaceFootnotes` and `\ifFBFrenchFootnotes` which are set by options of `\frenchbsetup{}` (see section 1.18). Notice that the layout of footnotes *does not depend* on the current language (just think of two footnotes on the same page looking different because one was called in a French part, the other one in English!).

When `\ifFBAutoSpaceFootnotes` is true, `\@footnotemark` (whose definition is saved at the `\begin{document}` in order to include any customisation that packages might have done) is redefined to add a thin space before the number or symbol calling a footnote (any space typed in is removed first). This has no effect on the layout of the footnote itself.

```

739 \AtBeginDocument{\@ifpackageloaded{bigfoot}%
740                 {\FBWarning{frenchb.ldf}%
741                  {bigfoot package in use.\MessageBreak
742                   frenchb will NOT customise footnotes;\MessageBreak
743                   reported}}}%
744                 {\let\@footnotemarkORI\@footnotemark
745                  \def\@footnotemarkFB{\leavevmode\unskip\unkern
746                                       \,\@footnotemarkORI}%
747                  \ifFBAutoSpaceFootnotes
748                    \let\@footnotemark\@footnotemarkFB
749                  \fi}%
750                 }

```

We then define `\@makefntextFB`, a variant of `\@makefntext` which is responsible for the layout of footnotes, to match the specifications of the French ‘Imprimerie Nationale’: footnotes will be indented by `\parindentFFN`, numbers (if any) typeset on the baseline (instead of superscripts) and followed by a dot and an half quad space. Whenever symbols are used to number footnotes (as in `\thanks` for instance), we switch back to the standard layout (the French layout of footnotes is meant for footnotes numbered by Arabic or Roman digits).

The value of `\parindentFFN` will be redefined at the `\begin{document}`, as the maximum of `\parindent` and 1.5em *unless* it has been set in the preamble (the weird value 10in is just for testing whether `\parindentFFN` has been set or not).

```

751 \newcommand*{\dotFFN}{.}
752 \newcommand*{\kernFFN}{\kern .5em}
753 \newdimen\parindentFFN
754 \parindentFFN=10in
755 \def\ftnISSymbol{\@fnsymbol\c@footnote}
756 \long\def\@makefntextFB#1{\ifx\thefootnote\ftnISSymbol
757                          \@makefntextORI{#1}%
758                          \else
759                          \parindent=\parindentFFN
760                          \rule{\z@}{\footnotesep}
761                          \setbox\@tempboxa\hbox{\@thefnmark}%
762                          \ifdim\wd\@tempboxa>\z@
763                          \llap{\@thefnmark}\dotFFN\kernFFN
764                          \fi #1

```

```
765 \fi}%
```

We save the standard definition of `\@makefntext` at the `\begin{document}`, and then redefine `\@makefntext` according to the value of flag `\ifFBFrenchFootnotes` (true or false).

```
766 \AtBeginDocument{\@ifpackageloaded{bigfoot}{}%
767     {\ifdim\parindentFFN<10in
768     \else
769     \parindentFFN=\parindent
770     \ifdim\parindentFFN<1.5em\parindentFFN=1.5em\fi
771     \fi
772     \let\@makefntextORI\@makefntext
773     \long\def\@makefntext#1{%
774     \ifFBFrenchFootnotes
775     \@makefntextFB{#1}%
776     \else
777     \@makefntextORI{#1}%
778     \fi}%
779     }%
780 }
```

For compatibility reasons, we provide definitions for the commands dealing with the layout of footnotes in **frenchb** version 1.6. `\frenchbsetup{}` (see in section 1.18) should be preferred for setting these options. `\StandardFootnotes` may still be used locally (in minipages for instance), that’s why the test `\ifFBFrenchFootnotes` is done inside `\@makefntext`.

```
781 \newcommand*\AddThinSpaceBeforeFootnotes{\FBAutoSpaceFootnotestruer}
782 \newcommand*\FrenchFootnotes{\FBFrenchFootnotestruer}
783 \newcommand*\StandardFootnotes{\FBFrenchFootnotesfalse}
```

## 1.16 Global layout

In multilingual documents, some typographic rules must depend on the current language (e.g., hyphenation, typesetting of numbers, spacing before double punctuation...), others should, IMHO, be kept global to the document: especially the layout of lists (see 1.13) and footnotes (see 1.15), and the indentation of the first paragraph of sections (see 1.14).

From version 2.2 on, if **frenchb** is **babel**’s “main language” (i.e. last language option at **babel**’s loading), **frenchb** customises the layout (i.e. lists, indentation of the first paragraphs of sections and footnotes) in the whole document regardless the current language. On the other hand, if **frenchb** is *not* **babel**’s “main language”, it leaves the layout unchanged both in French and in other languages.

`\FrenchLayout`    The former commands `\FrenchLayout` and `\StandardLayout` are kept for compatibility reasons but should no longer be used.

`\StandardLayout`

```
784 \newcommand*\FrenchLayout{%
785     \FBGlobalLayoutFrenchtrue
786     \PackageWarning{frenchb.ldf}%
787     {\protect\FrenchLayout\space is obsolete. Please use\MessageBreak
788     \protect\frenchbsetup{GlobalLayoutFrench} instead.}%
789 }
790 \newcommand*\StandardLayout{%
```



```

791 \FBReduceListSpacingfalse
792 \FBCompactItemizefalse
793 \FBStandardItemLabelstrue
794 \FBIndentFirstfalse
795 \FBFrenchFootnotesfalse
796 \FBAutoSpaceFootnotesfalse
797 \PackageWarning{frenchb.ldf}%
798     {\protect\StandardLayout\space is obsolete. Please use\MessageBreak
799     \protect\frenchbsetup{StandardLayout} instead.}%
800 }
801 \@onlypreamble\FrenchLayout
802 \@onlypreamble\StandardLayout

```

## 1.17 Dots...

`\FBtextellipsis`  $\text{\LaTeX 2}_\epsilon$ 's standard definition of `\dots` in text-mode is `\textellipsis` which includes a `\kern` at the end; this space is not wanted in some cases (before a closing brace for instance) and `\kern` breaks hyphenation of the next word. We define `\FBtextellipsis` for French (in  $\text{\LaTeX 2}_\epsilon$  only).

The `\if` construction in the  $\text{\LaTeX 2}_\epsilon$  definition of `\dots` doesn't allow the use of `xspace` (`xspace` is always followed by a `\fi`), so we use the AMS- $\text{\LaTeX}$  construction of `\dots`; this has to be done 'AtBeginDocument' not to be overwritten when `amsmath.sty` is loaded after `babel`.

LY1 has a ready made character for `\textellipsis`, it should be used in French too. The same is true for Unicode fonts in use with XeTeX and LuaTeX.

```

803 \ifFBunicode
804   \let\FBtextellipsis\textellipsis
805 \else
806   \DeclareTextSymbol{\FBtextellipsis}{LY1}{133}
807   \DeclareTextCommandDefault{\FBtextellipsis}{%
808     .\kern\fontdimen3\font.\kern\fontdimen3\font.\xspace}
809 \fi

```

`\Mdots@` and `\Tdots@ORI` hold the definitions of `\dots` in Math and Text mode. They default to those of `amsmath-2.0`, and will revert to standard  $\text{\LaTeX}$  definitions 'AtBeginDocument', if `amsmath` has not been loaded. `\Mdots@` doesn't change when switching from/to French, while `\Tdots@` is `\FBtextellipsis` in French and `\Tdots@ORI` otherwise.

```

810 \newcommand*{\Tdots@ORI}{\@xp\textellipsis}
811 \newcommand*{\Tdots@}{\Tdots@ORI}
812 \newcommand*{\Mdots@}{\@xp\Mdots@}
813 \AtBeginDocument{\DeclareRobustCommand*{\dots}{\relax
814                   \csname\ifmmode M\else T\fi dots@\endcsname}%
815                   \@ifundefined{@xp}{\let\@xp\relax}{}%
816                   \@ifundefined{mdots@}{\let\Tdots@ORI\textellipsis
817                                         \let\Mdots@\mathellipsis
818                                         \let\mdots@\undefined}{}}
819 \def\bbl@frenchdots{\let\Tdots@\FBtextellipsis}
820 \def\bbl@nonfrenchdots{\let\Tdots@\Tdots@ORI}
821 \addto\extrasfrench{\bbl@frenchdots}
822 \addto\noextrasfrench{\bbl@nonfrenchdots}

```

## 1.18 Setup options: keyval stuff

We first define a collection of conditionals with their defaults (true or false).

```

823 \newif\ifFBStandardLayout           \FBStandardLayouttrue
824 \newif\ifFBGlobalLayoutFrench       \FBGlobalLayoutFrenchfalse
825 \newif\ifFBReduceListSpacing        \FBReduceListSpacingfalse
826 \newif\ifFBListOldLayout            \FBListOldLayouttrue
827 \newif\ifFBCompactItemize           \FBCompactItemizefalse
828 \newif\ifFBStandardItemizeEnv        \FBStandardItemizeEnvtrue
829 \newif\ifFBStandardEnumerateEnv      \FBStandardEnumerateEnvtrue
830 \newif\ifFBStandardItemLabels        \FBStandardItemLabelstrue
831 \newif\ifFBStandardLists             \FBStandardListstrue
832 \newif\ifFBIndentFirst               \FBIndentFirstfalse
833 \newif\ifBBFrenchFootnotes           \BBFrenchFootnotesfalse
834 \newif\ifFBAutoSpaceFootnotes        \FBAutoSpaceFootnotesfalse
835 \newif\ifFBOriginalTypewriter        \FBOriginalTypewriterfalse
836 \newif\ifFBThinColonSpace           \FBThinColonSpacefalse
837 \newif\ifFBThinSpaceInFrenchNumbers \FBThinSpaceInFrenchNumbersfalse
838 \newif\ifBBFrenchSuperscripts        \BBFrenchSuperscriptstrue
839 \newif\ifFBLowercaseSuperscripts     \FBLowercaseSuperscriptstrue
840 \newif\ifFBPartNameFull              \FBPartNameFulltrue
841 \newif\ifFBSuppressWarning           \FBSuppressWarningfalse
842 \newif\ifFBShowOptions               \FBShowOptionsfalse

```

The defaults values of these flags have been set so that `frenchb` does not change anything regarding the global layout. `\bbl@main@language`, set by the last option of `babel`, controls the global layout of the document. ‘AtEndOfPackage’ we check the main language in `\bbl@main@language`; if it is French, the values of some flags have to be changed to ensure a French looking layout for the whole document (even in parts written in languages other than French); the end-user will then be able to customise the values of all these flags with `\frenchbsetup{}`.

```

843 \def\FB@french{french}
844 \AtEndOfPackage{%
845   \ifx\bbl@main@language\FB@french
846     \FBReduceListSpacingtrue
847     \FBListOldLayoutfalse
848     \FBCompactItemizetrue
849     \FBStandardItemizeEnvfalse
850     \FBStandardEnumerateEnvfalse
851     \FBStandardItemLabelsfalse
852     \FBIndentFirsttrue
853     \BBFrenchFootnotetrue
854     \FBAutoSpaceFootnotetrue
855     \FBGlobalLayoutFrenchtrue
856   \fi
857 }

```

`\frenchbsetup` From version 2.0 on, all setup options are handled by *one* command `\frenchbsetup` using the keyval syntax. Let’s now define this command which reads and sets the options to be processed later (at `\begin{document}`) by `\FBprocess@options`. It can only be called in the preamble.

```

858 \newcommand*{\frenchbsetup}[1]{%
859   \setkeys{FB}{#1}%

```

```

860 }%
861 \@onlypreamble\frenchbsetup

frenchb being an option of babel, it cannot load a package (keyval) while
frenchb.ldf is read, so we defer the loading of keyval and the options setup
at the end of babel's loading.

StandardLayout resets the layout in French to the standard layout defined par
the LATEX class and packages loaded. It deals with lists, indentation of first para-
graphs of sections and footnotes. Other keys, entered after StandardLayout in
\frenchbsetup, can overrule some of the StandardLayout settings.

When French is the main language, GlobalLayoutFrench forces the layout in
French and (as far as possible) outside French to meet the French typographic
standards.

862 \AtEndOfPackage{%
863     \RequirePackage{keyval}%
864     \define@key{FB}{StandardLayout}[true]%
865         {\csname FBStandardLayout#1\endcsname
866         \ifFBStandardLayout
867             \FBReduceListSpacingfalse
868             \FBStandardItemizeEnvtrue
869             \FBStandardEnumerateEnvtrue
870             \FBStandardItemLabelstrue
871             \FBIndentFirstfalse
872             \FBFrenchFootnotesfalse
873             \FBAutoSpaceFootnotesfalse
874             \FBGlobalLayoutFrenchfalse
875         \else
876             \FBReduceListSpacingtrue
877             \FBStandardItemizeEnvfalse
878             \FBStandardEnumerateEnvfalse
879             \FBStandardItemLabelsfalse
880             \FBIndentFirsttrue
881             \FBFrenchFootnotesttrue
882             \FBAutoSpaceFootnotesttrue
883         \fi}%
884     \define@key{FB}{GlobalLayoutFrench}[true]%
885         {\csname FBGlobalLayoutFrench#1\endcsname
886         \ifFBGlobalLayoutFrench
887             \ifx\bb1@main@language\FB@french
888                 \FBReduceListSpacingtrue
889                 \FBStandardItemizeEnvfalse
890                 \FBStandardEnumerateEnvfalse
891                 \FBStandardItemLabelsfalse
892                 \FBIndentFirsttrue
893                 \FBFrenchFootnotesttrue
894                 \FBAutoSpaceFootnotesttrue
895             \else
896                 \PackageWarning{frenchb.ldf}%
897                     {Option 'GlobalLayoutFrench' skipped:
898                     \MessageBreak French is *not*
899                     babel's last option.\MessageBreak}%
900             \fi
901         \fi}%
902     \define@key{FB}{ReduceListSpacing}[true]%

```

```

903         {\csname FBReduceListSpacing#1\endcsname}%
904 \define@key{FB}{ListOldLayout}[true]%
905         {\csname FBListOldLayout#1\endcsname
906         \ifFBListOldLayout
907         \renewcommand*{\FrenchLabelItem}{\textendash}%
908         \fi}%
909 \define@key{FB}{CompactItemize}[true]%
910         {\csname FBCompactItemize#1\endcsname
911         \ifFBCompactItemize
912         \FBStandardItemizeEnvfalse
913         \FBStandardEnumerateEnvfalse
914         \else
915         \FBStandardItemizeEnvtrue
916         \FBStandardEnumerateEnvtrue
917         \fi}%
918 \define@key{FB}{StandardItemizeEnv}[true]%
919         {\csname FBStandardItemizeEnv#1\endcsname}%
920 \define@key{FB}{StandardEnumerateEnv}[true]%
921         {\csname FBStandardEnumerateEnv#1\endcsname}%
922 \define@key{FB}{StandardItemLabels}[true]%
923         {\csname FBStandardItemLabels#1\endcsname}%
924 \define@key{FB}{ItemLabels}{%
925     \renewcommand*{\FrenchLabelItem}{#1}}%
926 \define@key{FB}{ItemLabeli}{%
927     \renewcommand*{\Frlabelitemi}{#1}}%
928 \define@key{FB}{ItemLabelii}{%
929     \renewcommand*{\Frlabelitemii}{#1}}%
930 \define@key{FB}{ItemLabeliii}{%
931     \renewcommand*{\Frlabelitemiii}{#1}}%
932 \define@key{FB}{ItemLabeliv}{%
933     \renewcommand*{\Frlabelitemiv}{#1}}%
934 \define@key{FB}{StandardLists}[true]%
935         {\csname FBStandardLists#1\endcsname
936         \ifFBStandardLists
937         \FBReduceListSpacingfalse
938         \FBCompactItemizefalse
939         \FBStandardItemizeEnvtrue
940         \FBStandardEnumerateEnvtrue
941         \FBStandardItemLabelstrue
942         \else
943         \FBReduceListSpacingtrue
944         \FBCompactItemizetrue
945         \FBStandardItemizeEnvfalse
946         \FBStandardEnumerateEnvfalse
947         \FBStandardItemLabelsfalse
948         \fi}%
949 \define@key{FB}{IndentFirst}[true]%
950         {\csname FBIndentFirst#1\endcsname}%
951 \define@key{FB}{FrenchFootnotes}[true]%
952         {\csname FBFrenchFootnotes#1\endcsname}%
953 \define@key{FB}{AutoSpaceFootnotes}[true]%
954         {\csname FBAutoSpaceFootnotes#1\endcsname}%
955 \define@key{FB}{AutoSpacePunctuation}[true]%
956         {\csname FBAutoSpacePunctuation#1\endcsname}%

```

```

957 \define@key{FB}{OriginalTypewriter}[true]%
958     {\csname FBOriginalTypewriter#1\endcsname}%
959 \define@key{FB}{ThinColonSpace}[true]%
960     {\csname FBThinColonSpace#1\endcsname}%
961 \define@key{FB}{ThinSpaceInFrenchNumbers}[true]%
962     {\csname FBThinSpaceInFrenchNumbers#1\endcsname}%
963 \define@key{FB}{FrenchSuperscripts}[true]%
964     {\csname FBFrenchSuperscripts#1\endcsname}%
965 \define@key{FB}{LowercaseSuperscripts}[true]%
966     {\csname FBLowercaseSuperscripts#1\endcsname}%
967 \define@key{FB}{PartNameFull}[true]%
968     {\csname FBPartNameFull#1\endcsname}%
969 \define@key{FB}{SuppressWarning}[true]%
970     {\csname FBSuppressWarning#1\endcsname
971     \ifFBSuppressWarning
972         \renewcommand{\FBWarning}[2]{\relax}%
973     \else
974         \renewcommand{\FBWarning}[2]{%
975             \PackageWarning{##1}{##2}}%
976     \fi}
977 \define@key{FB}{ShowOptions}[true]%
978     {\csname FBShowOptions#1\endcsname}%

```

Inputting French quotes as *single characters* when they are available on the keyboard (through a compose key for instance) is more comfortable than typing `\og` and `\fg`. The purpose of the following code is to map the French quote characters to `\og\ignorespaces` and `{\fg}` respectively when the current language is French, and to `\guillemotleft` and `\guillemotright` otherwise (think of German quotes); thus correct unbreakable spaces will be added automatically to French quotes. The quote characters typed in depend on the input encoding, it can be single-byte (latin1, latin9, applemac,...) or multi-bytes (utf-8, utf8x). We first check whether XeTeX or LuaTeX engines are used, if not the `inputenc` package has to be loaded before the `\begin{document}` with the proper coding option, so we check if `\DeclareInputText` is defined.

```

979 \define@key{FB}{og}{%
980     \newcommand*{\FB@og}{%
981         \iflanguage{french}%
982             {\ifFBAutoSpaceGuill\FB@og\ignorespaces
983             \else\guillemotleft
984             \fi}%
985         {\guillemotleft}}%
986 \ifFBunicode

```

LuaTeX or XeTeX in use,

```

987 \ifFB@xetex@punct

```

`\XeTeXinterchartokenstate` is defined, we just need to set `\XeTeXcharclass` to `\FB@punctguillo` for the French opening quote (see subsection 1.7) and to switch `\ifFBog@addspace` to false, otherwise commands `\og` and `\fg` would produce a double space; the `\ifFBguillo@addspace` flag is needed when switching from `\ttfamily` back to `\sf` or `\rm`.

```

988     \XeTeXcharclass"00AB = \FB@punctguillo
989     \FBguillo@addspacetrue \FBog@addspacefalse
990 \else

```

then LuaTeX or an old XeTeX in use, the following trick for defining the active quote character is borrowed from `inputenc.dtx`.

```

991      \catcode'#1=\active
992      \bgroup
993      \uccode'\~'#1%
994      \uppercase{%
995      \egroup
996      \def~%
997      }{\FB@@og}%
998      \fi
999      \else

This is for conventional TeX engines:

1000      \AtBeginDocument
1001      {\@ifundefined{DeclareInputText}%
1002      {\PackageWarning{frenchb.ldf}%
1003      {Option 'og' requires package inputenc.\MessageBreak}%
1004      \let\DeclareInputText\undefined
1005      }%
1006      {\@ifundefined{uc@dcclc}%
1007      {\@ifundefined{DeclareUnicodeCharacter}%
1008      if \DeclareUnicodeCharacter is undefined, utf8 is not loaded...
1009      {\@ifundefined{DeclareUnicodeCharacter}%
1010      if \DeclareUnicodeCharacter is undefined, utf8 is not loaded either, we as-
1011      sume 8-bit character input encoding. Package MULEenc.sty (from CJK) defines
1012      \mule@def to map characters to control sequences.
1013      {\@tempcnta'#1\relax
1014      \@ifundefined{mule@def}%
1015      {\DeclareInputText{\the\@tempcnta}{\FB@@og}%
1016      \let\mule@def\undefined}%
1017      {\mule@def{11}{\FB@@og}}}%
1018      \let\DeclareUnicodeCharacter\undefined
1019      }%
1020      utf8 loaded, use \DeclareUnicodeCharacter,
1021      {\DeclareUnicodeCharacter{00AB}{\FB@@og}}%
1022      \let\uc@dcclc\undefined
1023      }%
1024      utf8x loaded, use \uc@dcclc,
1025      {\uc@dcclc{171}{default}{\FB@@og}}%
1026      }%
1027      }%
1028      \fi
1029      }%

Same code for the closing quote.

1030      \define@key{FB}{fg}{%
1031      \newcommand*{\FB@@fg}{%
1032      \iflanguage{french}%
1033      {\ifFBAutoSpaceGuill\FB@fg
1034      \else\guillemotright
1035      \fi}%
1036      {\guillemotright}}}%
1037      \ifFBunicode

```

```

1031     \ifFB@xetex@punct
1032       \XeTeXcharclass"00BB = \FB@punctguilf
1033       \FBguilf@addspacetrue \FBfg@addspacefalse
1034     \else
1035       \catcode'#1=\active
1036       \bgroup
1037       \uccode'\~'#1%
1038       \uppercase{%
1039       \egroup
1040       \def~%
1041       }{{\FB@@fg}}%
1042     \fi
1043   \else
1044     \AtBeginDocument
1045     {\@ifundefined{DeclareInputText}%
1046      {\PackageWarning{frenchb.ldf}%
1047       {Option 'fg' requires package inputenc.\MessageBreak}%
1048      \let\DeclareInputText\undefined
1049      }%
1050      {\@ifundefined{uc@dclc}%
1051       {\@ifundefined{DeclareUnicodeCharacter}%
1052        {\@tempcnta'#1\relax
1053         \@ifundefined{mule@def}%
1054         {\DeclareInputText{\the\@tempcnta}{{\FB@@fg}}%
1055          \let\mule@def\undefined
1056          }%
1057          {\mule@def{27}{{\FB@@fg}}}%
1058          \let\DeclareUnicodeCharacter\undefined
1059          }%
1060          {\DeclareUnicodeCharacter{00BB}{{\FB@@fg}}%
1061           }%
1062          \let\uc@dclc\undefined
1063          }%
1064          {\uc@dclc{187}{default}{{\FB@@fg}}}%
1065          }%
1066        }%
1067      }%
1068    }%
1069  }

```

`\FBprocess@options` `\FBprocess@options` processes the options, it is called *once* at `\begin{document}`.

```

1070 \newcommand*{\FBprocess@options}{%

```

Nothing has to be done here for `StandardLayout` and `StandardLists` (the involved flags have already been set in `\frenchbsetup{}` or before (at babel's `EndOfPackage`).

The next three options deal with the layout of lists in French.

`ReduceListSpacing` reduces the vertical spaces in lists in French (done by changing `\list` to `\listFB`). When `GlobalLayoutFrench` is true (the default), the same is done outside French except for languages that force a different setting.

```

1071   \ifFBReduceListSpacing
1072     \addto\extrasfrench{\let\list\listFB
1073                        \let\endlist\endlistFB}%
1074   \addto\noextrasfrench{\ifFBGlobalLayoutFrench

```

```

1075             \let\list\listFB
1076             \let\endlist\endlistFB
1077         \else
1078             \let\list\listORI
1079             \let\endlist\endlistORI
1080         \fi}%
1081 \else
1082     \addto\extrasfrench{\let\list\listORI
1083                       \let\endlist\endlistORI
1084                       }%
1085     \addto\noextrasfrench{\let\list\listORI
1086                       \let\endlist\endlistORI
1087                       }%
1088 \fi

```

When ListOldLayout=true, set StandardEnumerateEnv=true (before v. 2.6a enumerate was not redefined).

```

1089 \ifFBListOldLayout
1090     \FBStandardEnumerateEnvtrue
1091 \fi

```

To ensure compatibility with the enumitem package, when enumitem is loaded, force flags StandardItemizeEnv and StandardEnumerateEnv to true.

```

1092 \@ifpackageloaded{enumitem}{%
1093     \ifFBStandardItemizeEnv
1094     \else
1095         \FBWarning{frenchb.ldb}{%
1096             {Setting StandardItemizeEnv=true for\MessageBreak
1097             compatibility with enumitem package,\MessageBreak}%
1098         \FBStandardItemizeEnvtrue
1099     \fi
1100     \ifFBStandardEnumerateEnv
1101     \else
1102         \FBWarning{frenchb.ldb}{%
1103             {Setting StandardEnumerateEnv=true for\MessageBreak
1104             compatibility with enumitem package,\MessageBreak}%
1105         \FBStandardEnumerateEnvtrue
1106     \fi}%
1107 {}%

```

To ensure compatibility with the paralist package, when paralist is loaded; force flags StandardItemizeEnv and StandardEnumerateEnv to true.

```

1108 \@ifpackageloaded{paralist}{%
1109     \ifFBStandardItemizeEnv
1110     \else
1111         \FBWarning{frenchb.ldb}{%
1112             {Setting StandardItemizeEnv=true for\MessageBreak
1113             compatibility with paralist package,\MessageBreak}%
1114         \FBStandardItemizeEnvtrue
1115     \fi
1116     \ifFBStandardEnumerateEnv
1117     \else
1118         \FBWarning{frenchb.ldb}{%
1119             {Setting StandardEnumerateEnv=true for\MessageBreak
1120             compatibility with paralist package,\MessageBreak}%

```



```

1121     \FBStandardEnumerateEnvtrue
1122     \fi}%
1123     {}%

```

To ensure compatibility with the `enumerate` package, when `enumerate` is loaded, force flag `StandardEnumerateEnv` to true.

```

1124 \@ifpackageloaded{enumerate}{%
1125     \ifFBStandardEnumerateEnv
1126     \else
1127         \FBWarning{frenchb.1df}%
1128         {Setting StandardEnumerateEnv=true for\MessageBreak
1129          compatibility with enumerate package,\MessageBreak}%
1130     \FBStandardEnumerateEnvtrue
1131     \fi}%
1132     {}%

```

Let's now process flags `StandardItemizeEnv` and `StandardEnumerateEnv`. Mind that `StandardEnumerateEnv`'s value decides for both environments `enumerate` and `description`).

```

1133 \ifFBStandardItemizeEnv
1134     \addto\extrasfrench{\let\itemize\itemizeORI}%
1135     \addto\noextrasfrench{\let\itemize\itemizeORI}%
1136 \else
1137     \addto\extrasfrench{\let\itemize\itemizeFB}%
1138     \addto\noextrasfrench{\ifFBGlobalLayoutFrench
1139                             \let\itemize\itemizeFB
1140                             \else
1141                             \let\itemize\itemizeORI
1142                             \fi}%
1143 \fi
1144 \ifFBStandardEnumerateEnv
1145     \addto\extrasfrench{\let\enumerate\enumerateORI
1146                         \let\description\descriptionORI}%
1147     \addto\noextrasfrench{\let\enumerate\enumerateORI
1148                           \let\description\descriptionORI}%
1149 \else
1150     \addto\extrasfrench{\let\enumerate\enumerateFB
1151                         \let\description\descriptionFB}%
1152     \addto\noextrasfrench{\ifFBGlobalLayoutFrench
1153                             \let\enumerate\enumerateFB
1154                             \let\description\descriptionFB
1155                             \else
1156                             \let\enumerate\enumerateORI
1157                             \let\description\descriptionORI
1158                             \fi}%
1159 \fi

```

`StandardItemLabels` resets `labelitems` in French to their standard values set by the `LATEX` class and packages loaded. When `GlobalLayoutFrench` is true, `labelitems` are identical whether the current language is French or not.

```

1160 \ifFBStandardItemLabels
1161     \addto\extrasfrench{\bbl@nonfrenchlabelitems}%
1162     \addto\noextrasfrench{\bbl@nonfrenchlabelitems}%
1163 \else
1164     \addto\extrasfrench{\bbl@frenchlabelitems}%

```

```

1165     \addto\noextrasfrench{\ifFBGlobalLayoutFrench
1166                             \bbl@frenchlabelitems
1167                             \else
1168                             \bbl@nonfrenchlabelitems
1169                             \fi}%
1170 \fi

```

**IndentFirst** forces the first paragraphs of sections to be indented just like the other ones in French. When **GlobalLayoutFrench** is true, the same is done outside French except for languages that force a different setting. **\bbl@nonfrenchindent** has been designed to be smart with other languages (like Spanish) who also indent every first paragraphs of sections (see section 1.14).

```

1171 \ifFBIndentFirst
1172     \addto\extrasfrench{\bbl@frenchindent}%
1173     \addto\noextrasfrench{\ifFBGlobalLayoutFrench
1174                             \bbl@frenchindent
1175                             \else
1176                             \bbl@nonfrenchindent
1177                             \fi}%
1178 \else
1179     \addto\extrasfrench{\bbl@nonfrenchindent}%
1180     \addto\noextrasfrench{\bbl@nonfrenchindent}%
1181 \fi

```

The layout of footnotes is handled at the `\begin{document}` depending on the values of flags **FrenchFootnotes** and **AutoSpaceFootnotes** (see section 1.15), nothing has to be done here for footnotes.

**AutoSpacePunctuation** adds an unbreakable space (in French only) before the four active characters (.;!?) even if none has been typed before them.

```

1182 \ifBBAutoSpacePunctuation
1183     \autospace@beforeFDP
1184 \else
1185     \noautospace@beforeFDP
1186 \fi

```

When **OriginalTypewriter** is set to **false** (the default), **\ttfamily**, **\rmfamily** and **\sffamily** are redefined as **\ttfamilyFB**, **\rmfamilyFB** and **\sffamilyFB** respectively to prevent addition of automatic spaces before the four active characters in computer code.

```

1187 \ifFBOriginalTypewriter
1188 \else
1189     \let\ttfamily\ttfamilyFB
1190     \let\rmfamily\rmfamilyFB
1191     \let\sffamily\sffamilyFB
1192 \fi

```

**ThinColonSpace** changes the normal unbreakable space typeset in French before ‘.’ to a thin space.

```

1193 \ifFBThinColonSpace\renewcommand*{\Fcolonspace}{\Fthinspace}\fi

```

When true, **ThinSpaceInFrenchNumbers** redefines **numprint.sty**’s command **\npstylefrench** to set **\npthousandsep** to **\**, (**thinspace**) instead of **~** (default) . This option has no effect if package **numprint.sty** is not loaded with ‘**autolanguage**’. As old versions of **numprint.sty** did not define **\npstylefrench**, we have to provide this command.

```

1194 \@ifpackageloaded{numprint}%
1195 {\ifnprt@autolanguage
1196   \providecommand*\npstylefrench{}%
1197   \ifFBThinSpaceInFrenchNumbers
1198     \renewcommand*\npstylefrench{%
1199       \npthousandsep{,}%
1200       \npdecimalsign{,}%
1201       \npproductsign{\cdot}%
1202       \npunitseparator{,}%
1203       \npdegreeseparator{}%
1204       \nppercentseparator{\nprt@unitsep}%
1205     }%
1206   \else
1207     \renewcommand*\npstylefrench{%
1208       \npthousandsep{~}%
1209       \npdecimalsign{,}%
1210       \npproductsign{\cdot}%
1211       \npunitseparator{,}%
1212       \npdegreeseparator{}%
1213       \nppercentseparator{\nprt@unitsep}%
1214     }%
1215   \fi
1216   \npaddtolanguage{french}{french}%
1217   \fi}{}%

```

FrenchSuperscripts: if true `\up=\fup`, else `\up=\textsuperscript`. Anyway `\up*=\FB@up@fake`. The star-form `\up*{}` is provided for fonts that lack some superior letters: Adobe Jenson Pro and Utopia Expert have no “g superior” for instance.

```

1218 \ifFBFrenchSuperscripts
1219   \DeclareRobustCommand*\up{\@ifstar{\FB@up@fake}{\fup}}%
1220 \else
1221   \DeclareRobustCommand*\up{\@ifstar{\FB@up@fake}%
1222                                     {\textsuperscript}}%
1223 \fi

```

LowercaseSuperscripts: if true let `\FB@lc` be `\lowercase`, else `\FB@lc` is redefined to do nothing.

```

1224 \ifFBLowercaseSuperscripts
1225 \else
1226   \renewcommand*\FB@lc[1]{##1}%
1227 \fi

```

PartNameFull: if false, redefine `\partname`.

```

1228 \ifFBPartNameFull
1229 \else\addto\captionsfrench{\def\partname{Partie}}\fi

```

ShowOptions: if true, print the list of all options to the `.log` file.

```

1230 \ifFBShowOptions
1231   \GenericWarning{* }{%
1232     * **** List of possible options for frenchb ****\MessageBreak
1233     [Default values between brackets when frenchb is loaded *LAST*]%
1234     \MessageBreak
1235     ShowOptions=true [false]\MessageBreak
1236     StandardLayout=true [false]\MessageBreak

```

```

1237 GlobalLayoutFrench=false [true]\MessageBreak
1238 StandardLists=true [false]\MessageBreak
1239 IndentFirst=false [true]\MessageBreak
1240 ReduceListSpacing=false [true]\MessageBreak
1241 ListOldLayout=true [false]\MessageBreak
1242 StandardItemizeEnv=true [false]\MessageBreak
1243 StandardEnumerateEnv=true [false]\MessageBreak
1244 StandardItemLabels=true [false]\MessageBreak
1245 ItemLabels=\textendash, \textbullet,
1246 \protect\ding{43},... [\textendash]\MessageBreak
1247 ItemLabeli=\textendash, \textbullet,
1248 \protect\ding{43},... [\textendash]\MessageBreak
1249 ItemLabelii=\textendash, \textbullet,
1250 \protect\ding{43},... [\textendash]\MessageBreak
1251 ItemLabeliii=\textendash, \textbullet,
1252 \protect\ding{43},... [\textendash]\MessageBreak
1253 ItemLabeliv=\textendash, \textbullet,
1254 \protect\ding{43},... [\textendash]\MessageBreak
1255 FrenchFootnotes=false [true]\MessageBreak
1256 AutoSpaceFootnotes=false [true]\MessageBreak
1257 AutoSpacePunctuation=false [true]\MessageBreak
1258 OriginalTypewriter=true [false]\MessageBreak
1259 ThinColonSpace=true [false]\MessageBreak
1260 ThinSpaceInFrenchNumbers=true [false]\MessageBreak
1261 FrenchSuperscripts=false [true]\MessageBreak
1262 LowercaseSuperscripts=false [true]\MessageBreak
1263 PartNameFull=false [true]\MessageBreak
1264 SuppressWarning=true [false]\MessageBreak
1265 og= <left quote character>, fg= <right quote character>
1266 \MessageBreak
1267 *****
1268 \MessageBreak\protect\frenchbsetup{ShowOptions}}
1269 \fi
1270 }

```

At `\begin{document}` we save again the value of `\if@afterindent`, definitions of the list environments and the values of `labelitems` so that all changes made in the preamble are taken into account in languages other than French and in French with the `StandardLayout` option. We also have to provide an `\xspace` command in case the `xspace.sty` package is not loaded.

```

1271 \AtBeginDocument{%
1272   \ifx\@afterindentfalse\@afterindenttrue
1273     \let\@aifORI\@afterindenttrue
1274   \else \let\@aifORI\@afterindentfalse
1275   \fi
1276   \let\listORI\list
1277   \let\endlistORI\endlist
1278   \let\endlistFB\endlist
1279   \let\itemizeORI\itemize
1280   \let\enditemizeORI\endlist
1281   \let\enditemizeFB\endlist
1282   \let\enumerateORI\enumerate
1283   \let\endenumerateORI\endlist
1284   \let\endenumerateFB\endlist

```

```

1285 \let\descriptionORI\description
1286 \let\enddescriptionORI\endlist
1287 \let\enddescriptionFB\endlist
1288 \let\@ltiORI\labelitemi
1289 \let\@ltiiORI\labelitemii
1290 \let\@ltiiiORI\labelitemiii
1291 \let\@ltivORI\labelitemiv
1292 \providecommand*{\xspace}{\relax}%

```

Let's redefine some commands in hyperref's bookmarks.

```

1293 \@ifundefined{pdfstringdefDisableCommands}{%
1294   {\let\pdfstringdefDisableCommands\undefined}%
1295   {\pdfstringdefDisableCommands{%
1296     \let\up\relax
1297     \let\up\relax
1298     \let\degre\textdegree
1299     \let\degres\textdegree
1300     \def\ieme{e\xspace}%
1301     \def\iemes{es\xspace}%
1302     \def\ier{er\xspace}%
1303     \def\iers{ers\xspace}%
1304     \def\iere{re\xspace}%
1305     \def\ieres{res\xspace}%
1306     \def\FrenchEnumerate#1{#1\degre\space}%
1307     \def\FrenchPopularEnumerate#1{#1\degre)\space}%
1308     \def\No{N\degre\space}%
1309     \def\no{n\degre\space}%
1310     \def\Nos{N\degre\space}%
1311     \def\nos{n\degre\space}%
1312     \def\FB@og{\guillemotleft\space}%
1313     \def\FB@fg{\space\guillemotright}%
1314     \def\at{@}%
1315     \def\circonflexe{\string~}%
1316     \def\tild{\string~}%
1317     \let\bsc\textsc
1318     }%
1319   }%

```

It is time to process the options set with `\frenchoptions{}`. Then we need to execute either `\extrasfrench` and `\captionsfrench` or `\noextrasfrench` according to the current language at the `\begin{document}` (these three commands have been updated by `\FBprocess@options`). *But*, when French is the main language, `\extrasfrench` is executed *again* now (French has been switched on 'AtBeginDocument' some time before). This is harmless, except for `\bbl@frenchspacing` which will redefine `\bbl@nonfrenchspacing` to `\relax`, this will be wrong if the user switches to English. When French is *not* the main language, `\noextrasfrench` executes `\bbl@nonfrenchspacing` (`=\nonfrenchspacing`), thus eventually overwriting a `\frenchspacing` command issued by the main language (German, Spanish, etc.). So we have to define `\bbl@nonfrenchspacing` as `\relax` here and restore it's meaning afterwards.

```

1320 \FBprocess@options
1321 \let\bbl@nonfrenchspacingORI\bbl@nonfrenchspacing
1322 \let\bbl@nonfrenchspacing\relax
1323 \ifx\bbl@main@language\FB@french

```

```

1324     \extrasfrench\captionsfrench
1325     \else
1326         \noextrasfrench
1327     \fi
1328     \let\bbl@nonfrenchspacing\bbl@nonfrenchspacingORI

```

Some warnings are issued when output font encodings are not properly set. With XeLaTeX, `fontspec.sty` and `xunicode.sty` should be loaded; with (pdf)LaTeX, a warning is issued when OT1 encoding is in use at the `\begin{document}`. Mind that `\encodingdefault` is defined as ‘long’, defining `\FBOTone` with `\newcommand*` would fail!

```

1329     \ifFBXeTeX
1330         \@ifundefined{DeclareUTFcharacter}%
1331             {\PackageWarning{frenchb.ldb}%
1332              {Add \protect\usepackage{xltextra} to the\MessageBreak
1333               preamble of your document,}%
1334              \let\DeclareUTFcharacter\undefined}%
1335         {}%
1336     \else
1337         \begingroup \newcommand{\FBOTone}{OT1}%
1338         \ifx\encodingdefault\FBOTone
1339             \PackageWarning{frenchb.ldb}%
1340             {OT1 encoding should not be used for French.
1341              \MessageBreak
1342              Add \protect\usepackage[T1]{fontenc} to the
1343              preamble\MessageBreak of your document,}%
1344         \fi
1345     \endgroup
1346 \fi
1347 }

```

## 1.19 Clean up and exit

Load `frenchb.cfg` (should do nothing, just for compatibility).

```
1348 \loadlocalcfg{frenchb}
```

Final cleaning. The macro `\ldf@quit` takes care for setting the main language to be switched on at `\begin{document}` and resetting the category code of `@` to its original value. The config file searched for has to be `frenchb.cfg`, and `\CurrentOption` has been set to ‘french’, so `\ldf@finish\CurrentOption` cannot be used: we first load `frenchb.cfg`, then call `\ldf@quit\CurrentOption`.

```

1349 \FBclean@on@exit
1350 \ldf@quit\CurrentOption

```

## Change History

v2.0	<p>General: <code>\parindentFFN</code> not changed if already defined (required by JA for cah-gut.cls). 31</p> <p>Added warning for OT1 encoding. 44</p> <p><code>AtBeginDocument</code>, save again the definitions of the ‘list’ and ‘itemize’ environments and the values of labelitems. As of frenchb v. 1.6, ‘ORI’ values were set when reading frenchb.ldf, later changes were ignored. 44</p> <p>Footnotes are now printed by default ‘à la française’ for the whole document. 31</p> <p>New command <code>\frenchbsetup</code> added for global customisation. 34</p> <p><code>\bsc</code>: <code>\hbox</code> dropped, replaced by <code>\kern0pt</code>. 21</p> <p><code>\captionsfrench</code>: ‘Fig.’ changed to ‘Figure’ and ‘Tab.’ to ‘Table’. 25</p> <p>Set <code>\CaptionSeparator</code> in <code>\extrasfrench</code> now instead of <code>\captionsfrench</code> because it has to be reset when leaving French. 25</p> <p><code>\datefrench</code>: 2 ‘<code>\relax</code>’ added in <code>\today</code>’s definition. 19</p> <p><code>\FBtextellipsis</code>: Added special case for LY1 encoding, see bug report from Bruno Voisin (2004/05/18). 33</p> <p><code>\nombre</code>: <code>\nombre</code> requires now numprint.sty. 24</p>	v2.0d	<p>General: Options <code>og</code> and <code>fg</code> changed: limit the definition to French so that quote characters can be used in German. 34</p>
v2.0b	<p>General: Footnotes: Just do nothing (except warning) when the bigfoot package is loaded. 31</p>	v2.0e	<p>General: New option: StandardLists. 34</p>
v2.0c	<p>General: <code>\ThinSpaceInFrenchNumbers</code> added for compatibility with frenchb-1.x. 24</p> <p>Option <code>ThinSpaceInFrenchNumbers</code> added. 34</p> <p>There is no need to define here numprint’s command <code>\npstylefrench</code>, it will be re-defined ‘<code>AtBeginDocument</code>’ by <code>\FBprocess@options</code>. 24</p>	v2.0f	<p>General: <code>StandardLayout</code> option had no effect on lists. Test moved to <code>\FBprocess@options</code>. 34</p> <p>Two typos corrected in option <code>StandardLists</code>: <code>[false] → [true]</code> and <code>StandardLayout → StandardLists</code>. 34</p>
		v2.0g	<p>General: Revert previous change to <code>StandardLayout</code>. This option must set the three flags <code>\FBReduceListSpacingfalse</code>, <code>\FBCompactItemizefalse</code>, and <code>\FBStandardItemLabeltrue</code> instead of <code>\FBStandardListstrue</code>, so that later options can still change their value before executing <code>\FBprocess@options</code>. Same thing for option <code>StandardLists</code>. 34</p> <p><code>\StandardLayout</code>: Flag <code>\ifFBStandardLayout</code> not checked by <code>\FBprocess@options</code>, low-level flags have to be set one by one. 32</p>
		v2.1a	<p>General: Command <code>\fup</code> added to produce better superscripts than <code>\textsuperscript</code>. 19</p> <p>New option: <code>FrenchSuperscripts</code> to define <code>\up</code> as <code>\fup</code> or as <code>\textsuperscript</code>. 34</p> <p>New option: <code>LowercaseSuperscripts</code>. 34</p>
		v2.1b	<p><code>\datefrench</code>: <code>\today</code> changed (correction in 2.0 was wrong: <code>\today</code> was printed without spaces in toc). 19</p> <p>General: Disable some commands in bookmarks. 44</p>

<code>\fup</code> : Command <code>\fup</code> changed to use real superscripts from fourier v. 1.6. . . . .	19	and <code>\AutoSpaceBeforeFDP</code> now set the flag <code>\iffBAutoSpacePunctuation</code> accordingly (LaTeX only). . . .	16
v2.1c		In LaTeX, frenchb no longer adds spaces before ‘double punctuation’ characters in computer code. Suggested by Yannis Haralambous. . . . .	16
General: Added commands <code>\Nos</code> and <code>\nos</code> . . . . .	21	New option: <code>OriginalTypewriter</code> . Now frenchb switches to <code>\noautospace@beforeFDP</code> when a tt-font is in use. When <code>OriginalTypewriter</code> is set to true, frenchb behaves as in pre-2.3 versions. . . . .	34
<code>\degres</code> : Provide a temporary definition (hyperref safe) of <code>\degres</code> in case it has to be expanded in the preamble (by beamer’s <code>\title</code> command for instance). . . . .	23	<code>\fup</code> : <code>\lowercase</code> changed to <code>\MakeLowercase</code> as the former doesn’t work for non ASCII characters in encodings like <code>aplemac</code> , <code>utf-8</code> , . . . . .	19
<code>\up</code> : Provide a temporary definition (hyperref safe) of <code>\up</code> in case it has to be expanded in the preamble (by beamer’s <code>\title</code> command for instance). . . . .	19	v2.3b	
v2.1d		General: New commands <code>\dotFFN</code> and <code>\kernFFN</code> for more flexibility (suggested by JA). . . . .	31
General: Argument of <code>\ProvidesLanguage</code> changed above from ‘french’ to ‘frenchb’ (otherwise <code>\listfiles</code> prints no date/version information). The real name of current language (french) as to be corrected before calling <code>\LdfInit</code> . . . . .	9	v2.3c	
Avoid warning “ <code>\end</code> occurred when <code>\ifx</code> ... incomplete” with LaTeX-2.09. . . . .	9	General: Commands <code>\ttfamily</code> , <code>\rmfamily</code> and <code>\sffamily</code> have to be robust. Bug introduced in 2.3a, pointed out by Manuel Pégourié-Gonnard. . . . .	16
v2.2a		v2.3d	
General: The global layout of the document is no longer changed when frenchb is not the last option of babel ( <code>\bbl@main@language</code> ). Suggested by Ulrike Fischer. . . .	34	<code>\bbl@nonfrenchindent</code> : Bug correction: previous versions of frenchb set the flag <code>\if@afterindent</code> to false outside French which is correct for English but wrong for some languages like Spanish. Pointed out by Juan José Torrens. . . .	30
Values of flags <code>\ifBReduceListSpacing</code> , <code>\ifBCompactItemize</code> , <code>\ifBStandardItemLabels</code> , <code>\ifBIndentFirst</code> , <code>\ifBFrenchFootnotes</code> , <code>\ifBAutoSpaceFootnotes</code> changed: default now means ‘StandardLayout’, it will be changed to ‘FrenchLayout’ AtEndOfPackage only if french is <code>\bbl@main@language</code> . . . .	34	<code>\frenchbsetup</code> : Warning added to <code>\GlobalLayoutFrench</code> when French is not the main language. . . . .	35
When frenchb is babel’s last option, French becomes the document’s main language, so <code>GlobalLayoutFrench</code> applies. . . .	34	v2.3e	
<code>\fup</code> : <code>\newif</code> and <code>\newdimen</code> moved before <code>\ifLaTeXe</code> to avoid an error with plainTeX. . . . .	19	General: Execute <code>\AutoSpaceBeforeFDP</code> also in LaTeX to define <code>\FDP@colonspace</code> : needed for <code>tex4ht</code> , pointed out by MPG. .	16
v2.3a		v2.4a	
General: <code>\NoAutoSpaceBeforeFDP</code>		General: <code>\PackageWarning</code> changed to <code>\FBWarning</code> (when bigfoot package in use). . . . .	31
		New option <code>SuppressWarning</code> . . . .	34



	<code>\CaptionSeparator: \PackageWarning</code> changed to <code>\FBWarning</code> (in case <code>\@makecaption</code> has been cus- tomized). <code>\FBWarning</code> is de- fined as <code>\PackageWarning</code> by default but can be made silent using <code>\frenchbsetup</code> , (sug- gested by MPG). . . . .	26		
	<code>\ifXeTeX</code> : Added a new ‘if’ <code>\FBunicode</code> and some <code>\lcode</code> definitions to <code>\extrasfrench</code> and <code>\noextrasfrench</code> . . . . .	11		
v2.4b	<code>\FBprocessoptions</code> : false when the <code>enumitem</code> package is loaded (ensures compatibility). . . . .	40		
v2.4c	General: In <code>\ttfamilyFB</code> , also cancel automatic spaces in- side French guillemets en- tered as characters (see <code>\frenchbsetup</code> ). . . . .	16		
	<code>\frenchbsetup</code> : In <code>\ttfamilyFB</code> , also cancel automatic spaces inside French guillemets coded as characters (see <code>\frenchbsetup</code> ). . . . .	37		
v2.4d	<code>\up</code> : Command <code>\up</code> defined with <code>\providecommand</code> instead of <code>\newcommand</code> as <code>\up</code> may be defined elsewhere ( <code>catalan.ldf</code> ). Bug pointed out by Felip Manyé i Ballester. . . . .	19		
v2.5a	General: <code>\og</code> and <code>\fg</code> do not print correctly in English when us- ing XeTeX or LuaTeX, fixed by using <code>\textquotedblleft</code> and <code>\textquotedblright</code> de- fined above. . . . .	18		
	Define <code>\bbl@nonfrenchspacing</code> locally as <code>\relax</code> , other- wise the <code>\bbl@frenchspacing</code> command included in <code>ger- manb.ldf</code> is overwritten here by <code>\noextrasfrench</code> . Bug pointed out by Ulrike Fischer. . . . .	45		
	Define <code>\Fthinspace</code> for those who want to customise the width of the space before ; and ?. . . . .	11		
	New command <code>\NoAutoSpacing</code> , suggested by MPG. . . . .	17		
				New definitions needed for Xe- TeX/LuaTeX to properly print some dates and captions: using c.s. like <code>\’e</code> do not work with XeTeX (OK with XeLaTeX). . . . .
				Punctuation is no longer made ac- tive with XeTeX-based engines. . . . .
				Test <code>\@ifundefined</code> leaves the tested control sequence de- fined as <code>\relax</code> when TRUE. Changed <code>\relax</code> to <code>\undefined</code> when testing <code>\numprint</code> . . . . .
				Test <code>\@ifundefined</code> leaves the tested control sequence defined as <code>\relax</code> when TRUE. Changed <code>\relax</code> to <code>\undefined</code> when testing <code>\pdfstringdefDisableCommands</code> <code>AtBeginDocument</code> . . . . .
				<code>\captionsfrench</code> : <code>\emph</code> deleted in <code>\seename</code> and <code>\alsoname</code> to match what is done for the other languages. Suggested by Marc Baudoin. . . . .
				Replaced <code>\’e</code> , <code>\’e</code> and <code>\’a</code> by c.s. to work with XeTeX. . . . .
				<code>\datefrench</code> : Replaced <code>\’e</code> and <code>\u</code> by c.s. to work with XeTeX and LuaTeX. . . . .
				<code>\frenchbsetup</code> : Test <code>\@ifundefined</code> leaves the tested control se- quence defined as <code>\relax</code> when TRUE. Changed <code>\relax</code> to <code>\undefined</code> when testing <code>\XeTeXrevision</code> , <code>\DeclareInputText</code> , <code>\uc@cllc</code> , <code>\DeclareUnicodeCharacter</code> , <code>\mule@def</code> in <code>\og</code> and <code>\fg</code> . . . . .
				<code>\textquoteddblright</code> : Change <code>\guillemotleft</code> and <code>\guillemotright</code> definitions for Unicode anf provide definitions for <code>\textquotedblleft</code> and <code>\textquotedblright</code> . Insures correct printing of quotes by <code>\og</code> and <code>\fg</code> in French and out- side. . . . .
				<code>\up</code> : Test <code>\@ifundefined</code> leaves the tested control sequence defined as <code>\relax</code> when TRUE. Changed <code>\relax</code> to <code>\undefined</code> when testing <code>\realsuperscript</code> . . . . .

v2.5b	General: <code>\captionsfrench</code> will be executed ‘AtBeginDocument’ after <code>\FBprocess@options</code> , no need to add it here. . . . . 25	<code>\at</code> , <code>\circonflexe</code> , <code>\tild</code> , <code>\boi</code> and <code>\degre</code> for Unicode based engines. . . . . 22
	Do not use the test <code>\iflanguage{french}</code> to check whether French is the main language or not, as it might be erroneously positive when English is the main language and no hyphenation patterns are available for French. In this case <code>\l@french</code> and <code>\l@english</code> are 0. Pointed out by Günter Milde. . . . . 34, 45	<code>\FBtextellipsis</code> : Unicode fonts also provide a ready made character for <code>\textellipsis</code> , let’s just use it (reported by Maxime Chupin, 2011/06/04). . . . . 33
	<code>\frenchbsetup</code> : Do not use the test <code>\iflanguage{french}</code> to check whether French is the main language or not, as it might be erroneously positive when English is the main language and no hyphenation patterns are available for French. In this case <code>\l@french</code> and <code>\l@english</code> are 0. Pointed out by Günter Milde. . . . . 35	v2.5g
		General: <code>\FBguill@spacing</code> needs to be skipped in commands <code>\FB@og</code> and <code>\FB@fg</code> only when XeTeX’s “inter char” mechanism is triggered for quotes, see <code>\frenchbsetup</code> . . . . . 18
v2.5c	<code>\frenchbsetup</code> : The code meant for XeTeX also works for LuaTeX, we just need to change the test. 37	Add four <code>\newif</code> to control spacing of quotes (characters and control sequences). . . . . 12
		Redefine <code>\degre</code> , <code>\degres</code> <code>\at</code> <code>\circonflexe</code> and <code>\tild</code> for bookmarks. Add <code>\fup</code> also. . . 45
v2.5d	General: Moved the <code>\newcount</code> command outside <code>\ifFB@xetex@punct</code> ... <code>\fi</code> (it broke Plain formats). 12	Skip the message in Plain, <code>\PackageInfo</code> undefined for Plain formats. . . . . 12
	<code>\ifFXeTeX</code> : Added two new ‘if’ <code>\FBXeTeX</code> and <code>\FBLuaTeX</code> as XeTeX and behave differently regarding the status of the French “apostrophe”. . . . . 11	Switch flags <code>\iffBog@addspace</code> and <code>\iffBfg@addspace</code> to true, otherwise <code>\og</code> and <code>\fg</code> provide no spacing in XeLaTeX with options ‘og’ and ‘fg’ activated in <code>\frenchbsetup</code> (bug v. 2.5a-f). 17
		<code>\FB@xetex@punct@french</code> : XeTeX-charclass(es) for French quotes will be set to <code>\FB@punctguillo</code> and <code>\FB@punctguilf</code> by options ‘og’ and ‘fg’ in <code>\frenchbsetup</code> . French quotes should behave as normal characters by default in XeLaTeX as in LaTeX. . . . . 13
v2.5e	General: <code>\pdfstringdefDisableCommands</code> should redefine <code>\FB@og</code> and <code>\FB@fg</code> instead of <code>\og</code> and <code>\fg</code> so that it works also when quotes are entered as characters. Reported by Sébastien Gouezel. . . . . 44	<code>\frenchbsetup</code> : When <code>\iffB@xetex@punct</code> is true, ‘og’ and ‘fg’ options now set XeTeXcharclasses of these characters to <code>\FB@punctguillo</code> and <code>\FB@punctguilf</code> . Otherwise French quotes behave as normal characters (their XeTeXcharclass is 0). . . . . 37
	The redefinition of <code>\pdfstringdefDisableCommands</code> from <code>\relax</code> to <code>\undefined</code> was misplaced. Reported by Sébastien Gouezel. . . . . 44	<code>\NoAutoSpacing</code> : Switch flags <code>\iffBog@addspace</code> and <code>\iffBfg@addspace</code> to true, otherwise <code>\og</code> and <code>\fg</code> provide no spacing in XeLaTeX with options ‘og’ and ‘fg’ activated in <code>\frenchbsetup</code> (bug v. 2.5a-f). 17
v2.5f	General: Changed definitions of	

