

# A Babel language definition file for French

## frenchb.dtx v3.1h, 2015/08/19

Daniel Flipo  
daniel.flipo@free.fr

## Contents

<b>1 The French language</b>	<b>2</b>
1.1 Basic interface	2
1.2 Customisation	4
1.2.1 <code>\frenchbsetup</code>	4
1.2.2 Captions	8
1.3 Hyphenation checks	9
1.4 Changes	9
<b>2 The code</b>	<b>13</b>
2.1 Initial setup	13
2.2 Punctuation	16
2.2.1 Punctuation with LuaTeX	17
2.2.2 Punctuation with XeTeX	24
2.2.3 Punctuation with standard (pdf)TeX	27
2.2.4 Punctuation switches common to all engines	28
2.3 Commands for French quotation marks	30
2.4 Date in French	34
2.5 Extra utilities	35
2.6 Formatting numbers	39
2.7 Caption names	40
2.8 Dots...	44
2.9 More checks about packages' loading order	45
2.10 Setup options: keyval stuff	46
2.11 French lists	59
2.12 French indentation of sections	63
2.13 Formatting footnotes	63
2.14 Clean up and exit	65
<b>3 Change History</b>	<b>66</b>

# 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 (May 1996) as part of babel-3.6beta. Version 2.0a was released in February 2007 and version 3.0a in February 2014.

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

L<sup>A</sup>T<sub>E</sub>X-2.09 is no longer supported. This new version (3.x) has been designed to be used only with L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> and Plain formats based on TeX, pdfTeX, LuaTeX or XeTeX engines.

Changes between version 2.0 and v3.1h are listed in subsection 1.4 p. 9.

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 ‘high punctuation’ (: ; ! ?) in French, others modify the general layout (i.e. layout of lists, footnotes, indentation of first paragraphs of sections) and should apply to the whole document.

`frenchb` takes account of babel’s *main language* defined as the *last* option at babel’s loading. When French is not babel’s main language, `frenchb` does not alter the general 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>2</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{}` (see section 1.2 p. 4);
3. vertical spacing in general L<sup>A</sup>T<sub>E</sub>X lists is shortened;
4. footnotes are displayed “à la française”.
5. the separator following the table or figure number in captions is printed as ‘-’ instead of ‘:’; for changing this see 1.2.2 p. 8.

Regarding local typography, the command `\selectlanguage{french}` switches to the French language<sup>3</sup>, with the following effects:

<sup>1</sup>The file described in this section has version number v3.1h and was last revised on 2015/08/19.

<sup>2</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 p. 4).

<sup>3</sup> `\selectlanguage{français}` and `\selectlanguage{frenchb}` are no longer supported.

1. French hyphenation patterns are made active;
2. ‘high punctuation’ characters (: ; ! ?) automatically add correct spacing in French; this is achieved using callbacks in Lua(La)TeX or ‘XeTeXinterchar’ mechanism in Xe(La)TeX; with TeX’82 and pdf(La)TeX these four characters are made active in the whole document;
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). For customisation of caption names see section 1.2.2 p. 8.
5. the space after `\dots` is removed in French.

Some commands are provided by 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~>>`: `\og` and `\fg` provide better horizontal spacing (controlled by `\FBguillspace`). If French quote characters are available on your keyboard, you can use them, to get proper spacing in L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> see option `og=«`, `fg=»` p. 8.

`\og` and `\fg` can be used outside French, they typeset then English quotes “ and ”.

A new command `\frquote{}` has been added in version 3.1 to enter French quotations. `\frquote{texte}` is equivalent to `\og texte \fg{}` for short quotations. For quotations spreading over more than one paragraph, `\frquote` will add at the beginning of every paragraph of the quotation either an opening French guillemet («), or a closing one (») depending on option `EveryParGuill=open` or `=close`, see p. 7.

`\frquote` is recommended to enter embedded quotations “à la française”, several variants are provided through options:

- with LuaTeX based engines, every line of the inner quotation will start with a French opening or closing guillemet (« or ») depending on option `EveryLineGuill=open` (default) or `=close` unless you explicitly set `EveryLineGuill=none`, then `\frquote{}` will behave as with non-LuaTeX engines;
- with all other engines, the inner quotation is surrounded by double quotes (“*texte*”) unless option `InnerGuillSingle=true`, then a) the inner quotation is printed as `< texte >` and b) if the inner quotation spreads over more than one paragraph, every paragraph included in the inner quotation starts with a `<` or a `>`, depending on option `EveryParGuill=open` or `close`.

A starred variant `\frquote*` is meant for inner quotations which end together with the outer one: using `\frquote*` for the inner quotation will print only one closing quote character (the outer one) as recommended by the French ‘Imprimerie Nationale’.

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>). All these commands take advantage of real superscript letters when they are available in the current font.
3. Family names should be typeset in small capitals and never be hyphenated, the macro `\bsc` (boxed small caps) does this, e.g., `L.\bsc{Lamport}` will print the same as `L.\mbox{\textsc{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)” ( $N^o$   $N^{os}$   $n^o$  and  $n^{os}$ ) 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 nobreak 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:  $[\theta, \ ]$ ,  $(x, \ y)$ . `\StandardMathComma` switches back to the standard behaviour of the comma.
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

Customisation of frenchb relies on command `\frenchbsetup{}`, options are entered using the `keyval` syntax. The command `\frenchbsetup{}` is to appear in the preamble only (after loading `babel`).

### 1.2.1 `\frenchbsetup{options}`

`\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 braces, 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 can be used to avoid conflicts with classes or packages which customise lists or footnotes.

`GlobalLayoutFrench=false (true*)` should no longer be used; it was intended to emulate, when French is the main language, what prior versions of `frenchb` (pre-2.2) did: lists, and first paragraphs of sections would 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`).

`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 the list environment.

`ListOldLayout=true (false)` ; starting with version 2.6a, the layout of lists has changed regarding left margins' sizes and default `itemize` label (`'—'` instead 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), it is replaced by the next two options.

`StandardItemizeEnv=true (false*)` ; `frenchb` redefines the `itemize` environment to suppress any vertical space between items of `itemize` lists in French and customises left margins. Setting this option to `false` reverts to the standard definition of `itemize`.

`StandardEnumerateEnv=true (false*)` ; starting with version 2.6 `frenchb` redefines the `enumerate` and `description` environments to make left margins match those of the French version of `itemize` lists. Setting this option to `false` reverts to the standard definition of `enumerate` and `description`.

`StandardItemLabels=true (false*)` when set to `true` this option prevents `frenchb` from changing the labels in `itemize` lists in French.

`ItemLabels=\textbullet, \textendash, \ding{43},...(\textemdash*)` ; when `StandardItemLabels=false` (the default), this option enables to choose the label used in French `itemize` lists for all levels. The next four options do the same but each one for a specific level only. Note that the example `\ding{43}` requires `\usepackage{pifont}`.

`ItemLabeli=\textbullet, \textendash, \ding{43},...(\textemdash*)`

`ItemLabelii=\textbullet, \textendash, \ding{43},...(\textemdash*)`

`ItemLabeliii=\textbullet, \textendash, \ding{43},..(\textemdash*)`

`ItemLabeliv=\textbullet, \textendash, \ding{43},...(\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 setting all four options `ReduceListSpacing=false`, `StandardItemizeEnv=true`, `StandardEnumerateEnv=true` and `StandardItemLabels=true`.

`IndentFirst=false (true*)` ; set this option to `false` if you do not want frenchb to force indentation of the first paragraph of sections. When French is the main language, this option applies to all languages.

`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). Two commands `\StandardFootnotes` and `\FrenchFootnotes` are available to change the layout of footnotes locally; `\StandardFootnotes` can help 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 documents written before 2008 without changes: 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 typeset nobreak spaces the width of which is either `\FBthinspace` (defaults to thin space) before ‘;’ ‘!’ ‘?’ or `\FBcolonspace` (defaults to `\space`) before ‘:’; the defaults follow the French ‘Imprimerie Nationale’s recommendations. This is convenient in most cases but can lead to addition of spurious spaces in URLs, in MS-DOS paths or in timetables (10:55), except if they are typed in `\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 use the provided `\NoAutoSpacing` command inside a group in case an unwanted space is added by frenchb (i.e. `{\NoAutoSpacing 10:55}`).

`ThinColonSpace=true (false)` changes the inter-word unbreakable space added before the colon ‘:’ to a thin space, so that the same amount of space is added before any of the four ‘high 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 classes do so), you could get “Première partie 1”, “Deuxième partie 2” in the toc; when this occurs, this option should be set to `false`, part titles will then be printed as “Partie I”, “Partie II”.

`CustomiseFigTabCaptions=false (true*)` ; when `false` the default separator (colon) is used instead of `\CaptionSeparator`. Anyway, frenchb makes sure that the colon will be typeset with proper preceding space in French.

`OldFigTabCaptions=true (false)` is to be used when figures’ and tables’ captions must be typeset as with pre 3.0 versions of frenchb (with `\CaptionSeparator` in French and colon otherwise). Intended for standard  $\LaTeX$  classes only.

`SmallCapsFigTabCaptions=false (true*)` ; when set to `false`, `\figurename` and `\tablename` will be printed in French captions as “Figure” and “Table” instead of being printed in small caps (the default).

`SuppressWarning=true (false)` ; can be turned to `true` if you are bored with frenchb’s warnings.

`INGuillSpace=true (false)` resets the dimensions of spaces after opening French quotes and before closing French quotes to the French ‘Imprimerie Nationale’ standards (inter-word space). frenchb’s default setting produces slightly narrower spaces with lesser stretchability.

`EveryParGuill=open, close, none (open)` ; sets whether an opening quote (`«`) or a closing one (`»`) or nothing should be printed by `\frquote{}` at the beginning of every paragraph in case of a level 1 (outer) quotation spreading over more than one paragraph. This option is also considered for level 2 (inner) quotations to decide between `<` and `>` when `InnerGuillSingle=true` (see below).

`EveryLineGuill=open, close, none (open in LuaTeX, none otherwise)` ; with engines other than LuaTeX this option is set to `none` which means that nothing will be printed at the beginning of every line of inner quotations, trying to set this option will issue a warning in the `.log` file.

With LuaTeX based engines, this option is set to `open` by default, it ensures that a ‘«’ followed by proper kern will be repeated at the beginning of every line in case an embedded (inner) quotation spreads over more than one line (provided that both outer and inner quotations are entered with `\frquote{}`). Set this option to `close` if you want a ‘»’ instead of a ‘«’.

`InnerGuillSingle=true` (`false`) ; if `InnerGuillSingle=false` (default), inner quotations entered with `\frquote{}` start with “ and end with ”. If `InnerGuillSingle=true`, < and > are used instead of British double quotes. Please note that this option only makes sense when `EveryLineGuill=none`.

`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 works with LuaLaTeX and XeLaTeX; with pdfLaTeX it requires `inputenc` to be loaded with a proper encoding: 8-bits encoding (`latin1`, `latin9`, `ansinew`, `applemac`,...) or multi-byte encoding (`utf8`, `utf8x`).

**Options’ order** – Please remember that options are read in the order they appear in the `\frenchbsetup{}` command. Someone wishing that frenchb leaves the layout of lists and footnotes untouched but caring for indentation of first paragraph of sections should choose

`\frenchbsetup{StandardLayout,IndentFirst}` to get the expected layout. The reverse order `\frenchbsetup{IndentFirst,StandardLayout}` would lead to option `IndentFirst` being overwritten by `StandardLayout`.

## 1.2.2 Captions

Caption names can be customised in French using the simplified syntax introduced by babel 3.9, for instance: `\def\frenchproofname{Preuve}`. The older syntax `\addto\captionsfrench{\def\proofname{Preuve}}` still works. Keep in mind that *only* french can be used to redefine captions, even if babel’s option was entered as `francais` or `frenchb`.

When French is the main language, by default (see below) frenchb changes the separator (colon) used in figures’ and tables’ captions *for all languages* to `\CaptionSeparator` which defaults to ‘ – ’ and can be redefined in the preamble with `\renewcommand*{\CaptionSeparator}{...}`.

When French is not the main language, the colon is preserved for all languages but frenchb makes sure that a proper space is typeset before it.

Three new options are provided: if `CustomiseFigTabCaptions` is set to `false` the colon will be used as separator in all languages, with a proper space before the colon in French. The second option, `OldFigTabCaptions`, can be set to `true` to print figures’ and tables’ captions as they were with versions pre 3.0 of frenchb (using `\CaptionSeparator` in French and colon in other languages); this option only makes sense with the standard L<sup>A</sup>T<sub>E</sub>X classes `article`, `report` and `book`. The last option, `SmallCapsFigTabCaptions`, can be set to `false` to typeset `\figurename` and `\tablename` in French as “Figure” and “Table” rather than in small caps (the default).



### 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 pdfLaTeX on 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.

### 1.4 Changes

#### What’s new in version 3.1?

New command \frquote{} meant to enter French quotations, especially long ones (spreading over several paragraphs) and/or embedded ones. see p. 3 for details.

#### What’s new in version 3.0?

Many deep changes lead me to step frenchb’s version number to 3.0a:

- babel 3.9 is required now to process frenchb.ldf, this change allows for cleaner definitions of dates and captions for the Unicode engines LuaTeX and XeTeX and also provides a simpler syntax for end-users, see section 1.2.2 p.8.
- \frenchbsetup{} options management has been completely reworked; two new options added.
- Canadian French didn't work as a normal babel's dialect, it should now; btw. the French language should now be loaded as french, *not as* frenchb or francais and preferably as a *global* option of \documentclass. Some tolerance still exists in v3.0, but do not rely on it.
- frenchb no longer loads frenchb.cfg: customisation should definitely be done using \frenchbsetup{} options.
- Description lists labels are now indented; set \listindentFB=0pt to get the former layout.
- The last but not least change affects the (recent) LuaTeX-based engines, (this means version 0.76 as included in TL2013 and up): active characters are no longer used in French for 'high punctuation'. Functionalities and user interface are unchanged.

Many thanks to Paul Isambert who provided the basis for the lua code (see his presentation at GUT'2010) and kindly reviewed my first drafts suggesting significant improvements.

Please note that this code, still experimental, is likely to change until LuaTeX itself has reached version 1.0.

Starting with version 3.0c, frenchb no longer customises lists with the beamer class and offers a new option (`INGuillSpace`) to follow French 'Imprimerie Nationale' recommendations regarding quotes' spacing.

### What's new in version 2.6?

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

frenchb is now compatible with the paralist package.

Regarding the layout of figures' and tables' captions, version 2.6c is now fully compatible with AMS and koma-script classes and with caption and floatrow packages. Starting with version 2.6c, the frenchb.cfg file is no longer generated from frenchb.dtx, but it is still loaded (if found) for backward compatibility.

### What's new in version 2.5?

The main change is that active characters are no longer used in French with (recent) XeTeX-based engines (they still are with 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 (adapted from the `polyglossia` package).

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 TeX-, LuaTeX- and XeTeX-based engines.

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

### What's new in version 2.4?

A new option `SuppressWarning` has been added (deactivated 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 XeTeX and 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 2.3?

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 2.2?

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 2.1?

A new command `\fup` is provided to typeset better looking superscripts; it was designed using ideas from Jacques André, Thierry Bouche and René Fritz,

thanks to all of them! Former command `\up` is now defined as `\fup`, an option `FrenchSuperscripts=false` is provided for backward compatibility.

### What's new in version 2.0?

Here is the list of all changes:

- Support for  $\LaTeX$ -2.09 and for  $\LaTeX 2_{\epsilon}$  in compatibility mode has been dropped. This version is meant for  $\LaTeX 2_{\epsilon}$  and Plain based formats (like `bpplain`).  $\LaTeX 2_{\epsilon}$  formats based on `miTeX` 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 only 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 tables 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).

## 2 The code

### 2.1 Initial setup

If frenchb.ldf was loaded with babel's options francais or frenchb, we make it behave as if french was specified. In Plain formats, @ catcode is not 'letter'.

```
1 \chardef\atcatcode=\catcode'\@
2 \catcode'\@=11\relax
3 \def\bb@tempa{francais}
4 \ifx\CurrentOption\bb@tempa
5   \let\l@francais\l@french
6   \def\captionsfrancais{\captionsfrench}
7   \def\datefrancais{\datefrench}
8   \def\extrasfrancais{\extrasfrench}
9   \def\noextrasfrancais{\extrasfrench}
10  \def\CurrentOption{french}
11 \fi
12 \def\bb@tempa{frenchb}
13 \ifx\CurrentOption\bb@tempa
14   \let\l@frenchb\l@french
15   \def\captionsfrenchb{\captionsfrench}
16   \def\datefrenchb{\datefrench}
17   \def\extrasfrenchb{\extrasfrench}
18   \def\noextrasfrenchb{\extrasfrench}
19   \def\CurrentOption{french}
20 \fi
21 \catcode'\@=\atcatcode \let\atcatcode\relax
```

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

```
22 \LdfInit\CurrentOption\captionsfrench
```

Make sure that \l@french is defined (possibly as 0). babel.def now (3.9i) defines \l@<language> also for eTeX, LuaTeX and XeTeX formats which set \lang@<language>.

```
23 \def\FB@nopatterns{%
24   \ifx\l@nohyphenation\undefined
25     \edef\bb@nulllanguage{\string\language=0}%
26     \adddialect\l@french0
27   \else
28     \adddialect\l@french\l@nohyphenation
29     \edef\bb@nulllanguage{\string\language=nohyphenation}%
30   \fi
31   \@nopatterns{French}}
32 \ifx\l@french\undefined
33   \FB@nopatterns
34 \fi
```

**\ifLaTeXe** No support is provided for late L<sup>A</sup>T<sub>E</sub>X-2.09: issue a warning and exit if L<sup>A</sup>T<sub>E</sub>X-2.09 is in use. Plain is still supported.

```

35 \newif\ifLaTeXe
36 \let\bbbl@tempa\relax
37 \ifx\magnification\@undefined
38   \ifx\@compatibilitytrue\@undefined
39     \PackageError{frenchb.ldf}
40       {LaTeX-2.09 format is no longer supported.\MessageBreak
41         Aborting here}
42       {Please upgrade to LaTeX2e!}
43     \let\bbbl@tempa\endinput
44   \else
45     \LaTeXetrue
46   \fi
47 \fi
48 \bbbl@tempa

```

Let's provide a substitute for `\PackageError`, `\PackageWarning` and `\PackageInfo` not defined in Plain:

```

49 \def\fb@error#1#2{%
50   \begingroup
51     \newlinechar='\^^J
52     \def\{\^^J(frenchb.ldf) }%
53     \errhelp{#2}\errmessage{\#\1}%
54   \endgroup}
55 \def\fb@warning#1{%
56   \begingroup
57     \newlinechar='\^^J
58     \def\{\^^J(frenchb.ldf) }%
59     \message{\#\1}%
60   \endgroup}
61 \def\fb@info#1{%
62   \begingroup
63     \newlinechar='\^^J
64     \def\{\^^J}%
65     \wlog{#1}%
66   \endgroup}

```

Quit if babel's version is less than 3.9i.

```

67 \let\bbbl@tempa\relax
68 \ifx\babeltags\@undefined
69   \let\bbbl@tempa\endinput
70   \ifLaTeXe
71     \PackageError{frenchb.ldf}
72       {frenchb requires babel v.3.9i.\MessageBreak
73         Aborting here}
74       {Please upgrade Babel!}
75   \else
76     \fb@error{frenchb requires babel v.3.9i.\
77       Aborting here}
78       {Please upgrade Babel!}
79   \fi
80 \fi

```

```
81 \bbl@tempa
```

frenchb.ldf can be loaded with options `canadien` or `acadian`, which both stand for Canadian French. Internally, `acadian` will be the name of the corresponding babel's dialect, so we set `\CurrentOption` to `acadian` in both cases. If no specific hyphenation patterns are available, Canadian French will use the French ones.

TODO: Canadian French hyphenation doesn't work with LuaTeX.

```
82 \ifx\l@acadian\@undefined
83   \ifx\l@canadien\@undefined
84     \adddialect\l@acadian\l@french
85     \adddialect\l@canadien\l@french
86   \else
87     \adddialect\l@acadian\l@canadien
88   \fi
89 \else
90   \adddialect\l@canadien\l@acadian
91 \fi
92 \def\bbl@tempa{canadien}
93 \ifx\CurrentOption\bbl@tempa
94   \def\captionscanadien{\captionacadian}
95   \def\datecanadien{\dateacadian}
96   \def\extrascanadien{\extrasacadian}
97   \def\noextrascanadien{\extrasacadian}
98   \def\CurrentOption{acadian}
99 \fi
```

French uses the standard values of `\lefthyphenmin` (2) and `\righthyphenmin` (3); let's provide their values though, as required by babel.

```
100 \expandafter\providehyphenmins\expandafter{\CurrentOption}{\tw@\thr@@}
```

**\ifFBUnicode** French hyphenation patterns are now coded in Unicode, see file `hyph-fr.tex`. XeTeX

**\ifBFLuaTeX** and LuaTeX engines require some extra code to deal with the French "apostrophe".

**\ifFBXeTeX** Let's define three new 'if': `\ifBFLuaTeX`, `\ifFBXeTeX` and `\ifFBUnicode` which will be true for XeTeX and LuaTeX engines and false for 8-bits engines.

We cannot rely on  $\varepsilon$ -TeX's `\ifdefined` at this stage, as it is not defined in Plain T<sub>E</sub>X format.

```
101 \newif\ifFBUnicode
102 \newif\ifBFLuaTeX
103 \newif\ifFBXeTeX
104 \begingroup\expandafter\expandafter\expandafter\endgroup
105 \expandafter\ifx\csname luatexversion\endcsname\relax
106 \else
107   \FBunicodetrue \BFLuaTeXtrue
108 \fi
109 \begingroup\expandafter\expandafter\expandafter\endgroup
110 \expandafter\ifx\csname XeTeXrevision\endcsname\relax
111 \else
112   \FBunicodetrue \FBXeTeXtrue
113 \fi
```

`\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.

The following code ensures correct hyphenation of words like `d’aventure`, `l’utopie`, with all TeX engines (XeTeX, LuaTeX, pdfTeX) using `hyph-fr.tex` patterns.

```

114 \@namedef{extras\CurrentOption}{%
115     \babel@savevariable{\lccode'\'}%
116     \ifFBunicode
117         \babel@savevariable{\lccode"2019}%
118         \lccode'\'"2019\lccode"2019="2019
119     \else
120         \lccode'\'"\'
121     \fi
122 }
123 \@namedef{noextras\CurrentOption}{}
```

Let’s define a handy command for adding stuff to `\extras\CurrentOption`, `\noextras\CurrentOption` or `\captions\CurrentOption` but first let’s save the value of `\CurrentOption` for later use in `\frenchbsetup{} (‘AfterEndOfPackage’, \CurrentOption will be lost).`

```

124 \let\FB@CurOpt\CurrentOption
125 \newcommand*{\FB@addto}[2]{%
126     \expandafter\addto\csname #1\FB@CurOpt\endcsname{#2}}
```

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

```

127 \FB@addto{extras}{\bbl@frenchspacing}
128 \FB@addto{noextras}{\bbl@nonfrenchspacing}
```

## 2.2 Punctuation

As long as no better solution is available, the ‘high punctuation’ characters (`;` `!` `?` and `:`) have to be made `\active` for an automatic control of the amount of space to be inserted before them. Both XeTeX and LuaTeX provide an alternative to active characters (‘XeTeXinterchar’ mechanism and LuaTeX’s callbacks).

With LuaTeX and XeTeX engines, `frenchb` handles French quotes together with ‘high punctuation’, a new conditional will be needed:

```

129 \newif\ifFBAutoSpaceGuill \FBAutoSpaceGuilltrue
```

`\ifFB@active@punct` Three internal flags are needed for the three different techniques used for ‘high punctuation’ management.

`\ifFB@xetex@punct` With XeTeX, starting with version 0.76, callbacks are used to get rid of active punctuation. With previous versions, ‘high punctuation’ characters remain active (see below).



```

130 \newif\ifFB@active@punct \FB@active@puncttrue
131 \newif\ifFB@luatex@punct
132 \ifBLaTeX
133 \ifnum\luatexversion>75
134 \FB@luatex@puncttrue\FB@active@punctfalse
135 \fi
136 \fi

```

For XeTeX, the availability of `\XeTeXinterchartokenstate` decides whether the ‘high punctuation’ characters (; ! ? and :) have to be made `\active` or not.

```

137 \newif\ifFB@xetex@punct
138 \begingroup\expandafter\expandafter\expandafter\endgroup
139 \expandafter\ifx\csname XeTeXinterchartokenstate\endcsname\relax
140 \else
141 \FB@xetex@puncttrue\FB@active@punctfalse
142 \fi

```

`\FBcolonspace` According to the I.N. specifications, the ‘:’ requires an inter-word space before it, the `\FBthinspace` other three require just a `\thinspace`. We define `\FBcolonspace` as `\space` (inter-word space) and `\FBthinspace` as `\thinspace` (both are user customisable). LuaTeX `\FBcolonskip` requires skips instead of commands, so we define `\FBcolonskip` and `\FBthinskip` to hold the specifications (width/stretch/shrink) of `\space` and `\thinspace` for the `lrm10` font; these parameters will be scaled for the current font by the `frenchb.lua` script (see how p. 19). `\FBcolonskip` and `\FBthinskip` are also user customisable.

```

143 \newcommand*{\FBcolonspace}{\space}
144 \newcommand*{\FBthinspace}{\hskip .16667em \relax}
145 \newskip\FBcolonskip
146 \FBcolonskip=3.33pt plus 1.665pt minus 1.11pt \relax
147 \newskip\FBthinskip
148 \FBthinskip=1.66672pt \relax

```

### 2.2.1 Punctuation with LuaTeX

The following part holds specific code for punctuation with modern LuaTeX engines (version  $\geq 0.76$ ).

We define two LuaTeX attributes to control spacing in French for ‘high punctuation’ and quotes, making sure that `\newluatexattribute` is defined.

```

149 \ifFB@luatex@punct
150 \ifLaTeXe
151 \AtEndOfPackage{%
152 \RequirePackage{luatexbase}%
153 \newluatexattribute\FB@addDPspace \FB@addDPspace=1 \relax
154 \newluatexattribute\FB@addGUILspace \FB@addGUILspace=0 \relax
155 }
156 \PackageInfo{frenchb.ldf}{No need for active punctuation characters%
157 \MessageBreak with this version of LuaTeX!%
158 \MessageBreak reported}
159 \else
160 \begingroup\expandafter\expandafter\expandafter\endgroup
161 \expandafter\ifx\csname newluatexattribute\endcsname\relax

```

```

162     \input luatexbase.sty
163     \fi
164     \newluatexattribute\FB@addDPspace   \FB@addDPspace=1 \relax
165     \newluatexattribute\FB@addGUILspace \FB@addGUILspace=0 \relax
166     \fb@info{No need for active punctuation characters\\
167             with this version of LuaTeX!}
168     \fi
169 \fi

```

frenchb.lua holds Lua code to deal with ‘high punctuation’ and quotes. This code is based on suggestions from Paul Isambert.

First we define two flags to control spacing before French ‘high punctuation’ (thin space or inter-word space).

```

170 local FB_punct_thin =
171   {[string.byte("!")] = true,
172    [string.byte("?")] = true,
173    [string.byte(";")] = true}
174 local FB_punct_thick =
175   {[string.byte(":")] = true}

```

Managing spacing after ‘«’ (U+00AB) and before ‘»’ (U+00BB) can be done by the way; we define two flags, FB\_punct\_left for characters requiring some space before them and FB\_punct\_right for ‘«’ which must be followed by some space. In case LuaTeX is used to output T1-encoded fonts instead of OpenType fonts, codes 0x13 and 0x14 have to be added for ‘«’ and ‘»’.

```

176 local FB_punct_left =
177   {[string.byte("!")] = true,
178    [string.byte("?")] = true,
179    [string.byte(";")] = true,
180    [string.byte(":")] = true,
181    [0x14]           = true,
182    [0xBB]          = true}
183 local FB_punct_right =
184   {[0x13]          = true,
185    [0xAB]          = true}

```

Two more flags will be needed to avoid spurious spaces in strings like !! ?? or (?)

```

186 local FB_punct_null =
187   {[string.byte("!")] = true,
188    [string.byte("?")] = true,
189    [string.byte("[")] = true,
190    [string.byte("(")] = true,

```

or if the user has typed a nobreak space U+00A0 or a nobreak thin space U+202F before a ‘high punctuation’ character: no space should be added by frenchb. Same is true inside French quotes.

```

191   [0xA0]           = true,
192   [0x202F]        = true}
193 local FB_guil_null =
194   {[0xA0]           = true,
195    [0x202F]        = true}

```

Local definitions for nodes:

```
196 local new_node      = node.new
197 local copy_node     = node.copy
198 local node_id       = node.id
199 local HLIST         = node_id("hlist")
200 local TEMP          = node_id("temp")
201 local KERN          = node_id("kern")
202 local GLUE          = node_id("glue")
203 local GSPEC         = node_id("glue_spec")
204 local GLYPH         = node_id("glyph")
205 local PENALTY       = node_id("penalty")
206 local nobreak       = new_node(PENALTY)
207 nobreak.penalty     = 10000
208 local insert_node_before = node.insert_before
209 local insert_node_after  = node.insert_after
210 local remove_node       = node.remove
```

Some variables to store `\FBthinskip`, `\FBcolonskip` and `\FBguillskip` (given for `\lrm10`); `width/stretch/shrink` are stored as fractions of `\fontdimen2`, `\fontdimen3` and `\fontdimen4` of `\lrm10` font respectively...

```
211 local thin10 = tex.skip['FBthinskip']
212 local thinwd = thin10.width/65536/3.33
213 local thinst = thin10.stretch/65536/1.665
214 local thinsh = thin10.shrink/65536/1.11
215 local coln10 = tex.skip['FBcolonskip']
216 local colnwd = coln10.width/65536/3.33
217 local colnst = coln10.stretch/65536/1.665
218 local colnsh = coln10.shrink/65536/1.11
219 local guil10 = tex.skip['FBguillskip']
220 local guilwd = guil10.width/65536/3.33
221 local guilst = guil10.stretch/65536/1.665
222 local guilsh = guil10.shrink/65536/1.11
```

and a function to scale them for the current font (beware of null values for `fid`, see `\nullfont` in TikZ):

```
223 local font_table = {}
224 local function new_glue_scaled (fid,width,stretch,shrink)
225   if fid > 0 then
226     local fp = font_table[fid]
227     if not fp then
228       font_table[fid] = font.getfont(fid).parameters
229       fp = font_table[fid]
230     end
231     local gl = new_node(GLUE,0)
232     local gl_spec = new_node(GSPEC)
233     gl_spec.width = width * fp.space
234     gl_spec.stretch = stretch * fp.space_stretch
235     gl_spec.shrink = shrink * fp.space_shrink
236     gl.spec = gl_spec
237     return gl
238   else
```

```

239     return nil
240 end
241 end

```

Let's catch LuaTeX attributes `\FB@addDPspace` and `\FB@addGUILspace`. Constant `FR=lang.id(french)` will be defined by command `\activate@luatexpunct`.

```

242 local addDPspace   = luatexbase.attributes['FB@addDPspace']
243 local addGUILspace = luatexbase.attributes['FB@addGUILspace']
244 local has_attribute = node.has_attribute

```

The following function will be added to kerning callback. It catches all nodes of type `GLYPH` in the list starting at `head` and checks the language attributes of the current glyph: nothing is done if the current language is not French and only specific punctuation characters (those for which `FB_punct_left` or `FB_punct_right` is true) need a special treatment. In French, local variables are defined to hold the properties of the current glyph (`item`) and of the previous one (`prev`) or the next one (`next`).

```

245 local function french_punctuation (head)
246   for item in node.traverse_id(GLYPH, head) do
247     local lang = item.lang
248     local char = item.char
249     local fid  = item.font
250     local SIG  = has_attribute(item, addGUILspace)
251     if lang == FR and FB_punct_left[char] and fid > 0 then
252       local prev = item.prev
253       local prev_id, prev_subtype, prev_char
254       if prev then
255         prev_id = prev.id
256         prev_subtype = prev.subtype
257         if prev_id == GLYPH then
258           prev_char = prev.char
259         end
260       end

```

If the previous item is a glue, check its natural width, only positive glues (actually glues  $> 1$  sp, for tabular 'l' columns) are to be replaced by a nobreakspace.

```

261     local glue = prev_id == GLUE and prev_subtype == 0
262     local glue_wd
263     if glue then
264       glue_spec = prev.spec
265       glue_wd = glue_spec.width
266     end
267     local realglue = glue and glue_wd > 1

```

For characters for which `FB_punct_thin` or `FB_punct_thick` is *true*, the amount of spacing to be typeset before them is controlled by `\FBthinskip` (`thinwd`, `thinstd`, `thinsh`) or `\FBcolonskip` (`colnwd`, `colnst`, `colnsh`) respectively. Two options: if a space has been typed in before (turned to *glue* in the node list), we remove the *glue* and add a nobreak penalty and the required *glue*. Otherwise (auto option), the penalty and the required *glue* are inserted if attribute `\FB@addDPspace` is set, unless one of these three condition is met: a) the previous character is part of type `FB_punct_null` (this avoids spurious spaces in strings like `( ! )` or `??`), b) a null glue

(actually glues  $\leq 1$  sp for tabulars) precedes the punctuation character, c) the punctuation character starts a paragraph or an `\hbox{}`.

```

268     if FB_punct_thin[char] or FB_punct_thick[char] then
269         local SBDP = has_attribute(item, addDPspace)
270         local auto = SBDP and SBDP > 0
271         if auto then
272             if (prev_char and FB_punct_null[prev_char]) or
273                 (glue and glue_wd <= 1) or
274                 (prev_id == HLIST and prev_subtype == 3) or
275                 (prev_id == TEMP) then
276                 auto = false
277             end
278         end
279         local fbglue
280         if FB_punct_thick[char] then
281             fbglue = new_glue_scaled(fid,colnwd,colnst,colnsh)
282         else
283             fbglue = new_glue_scaled(fid,thinwd,thinst,thinsh)
284         end
285         if realglue or auto then
286             if realglue then
287                 head = remove_node(head,prev,true)
288             end
289             insert_node_before(head, item, copy_node(nobreak))
290             insert_node_before(head, item, copy_node(fbglue))
291         end

```

Let's consider '»' now (the only remaining glyph of `FB_punct_left` class): we just have to remove any *glue* possibly preceding '»', then to insert the nobreak penalty and the proper *glue* (controlled by `\FBguillskip`). This is done only if French quotes have been 'activated' by options `og=«, fg=»` in `\frenchbsetup{}` and can be denied locally with `\NoAutoSpacing` (this is controlled by the SIG flag). If either a) the preceding glyph is member of `FB_guil_null`, or b) '»' is the first glyph of an `\hbox{}` or a paragraph, nothing is done, this is controlled by the `addgl` flag.

```

292     elseif SIG and SIG > 0 then
293         local addgl = (prev_char and not FB_guil_null[prev_char]) or
294             (not prev_char and
295                 prev_id ~= TEMP and
296                 not (prev_id == HLIST and prev_subtype == 3)
297             )

```

Correction for tabular 'c' (glue 0 plus 1 fil) and 'l' (glue 1sp) columns:

```

298     if glue and glue_wd <= 1 then
299         addgl = false
300     end
301     if addgl then
302         if glue then
303             head = remove_node(head,prev,true)
304         end
305         local fbglue = new_glue_scaled(fid,guilwd,guilst,guilsh)

```

```

306         insert_node_before(head, item, copy_node(nobreak))
307         insert_node_before(head, item, copy_node(fbg_lue))
308     end
309 end
310 end

```

Similarly, for '«' (unique member of the FB\_punct\_right class): unless either a) the next glyph is member of FB\_guil\_null, or b) '«' is the last glyph of an \hbox{} or a paragraph (then the addgl flag is false, nothing is done), we remove any *glue* possibly following it and insert first the proper *glue* then a nobreak penalty so that finally the penalty precedes the *glue*.

```

311     if lang == FR and FB_punct_right[char] and fid > 0
312         and SIG and SIG > 0 then
313         local next = item.next
314         local next_id, next_subtype, next_char, nextnext, kern_wd
315         if next then
316             next_id = next.id
317             next_subtype = next.subtype
318             if next_id == GLYPH then
319                 next_char = next.char

```

A kern0 might hide a glue, so look ahead if next is a kern (this occurs with « \texttt{a} »):

```

320             elseif next_id == KERN then
321                 kern_wd = next.kern
322                 if kern_wd == 0 then
323                     nextnext = next.next
324                     if nextnext then
325                         next = nextnext
326                         next_id = nextnext.id
327                         next_subtype = nextnext.subtype
328                         if next_id == GLYPH then
329                             next_char = nextnext.char
330                         end
331                     end
332                 end
333             end
334         end
335         local glue = next_id == GLUE and next_subtype == 0
336         if glue then
337             glue_spec = next.spec
338             glue_wd = glue_spec.width
339         end
340         local addgl = (next_char and not FB_guil_null[next_char]) or
341             (next and not next_char)

```

Correction for tabular 'c' columns. For 'r' columns, a final '«' character needs to be coded as \mbox{«} for proper spacing (\NoAutoSpacing is another option).

```

342         if glue and glue_wd == 0 then
343             addgl = false
344         end

```

```

345     if addgl then
346         if glue then
347             head = remove_node(head,next,true)
348         end
349         local fid = item.font
350         local fbg glue = new_glue_scaled(fid,guilwd,guilst,guilsh)
351         insert_node_after(head, item, copy_node(fbg glue))
352         insert_node_after(head, item, copy_node(nobreak))
353     end
354 end
355 end
356 return head
357 end
358 return french_punctuation

```

As a language tag is part of glyph nodes in LuaTeX, nothing needs to be added to `\extrasfrench` and `\noextrasfrench`; we will just redefine `\shorthandoff` and `\shorthandon` in French to issue a warning reminding the user that active characters are no longer used in French with recent LuaTeX engines.

```

359 \ifFB@luatex@punct
360 \newcommand*{\FB@luatex@punct@french}{%
361     \ifx\shorthandoffORI\undefined
362         \let\shorthandonORI\shorthandon
363         \let\shorthandoffORI\shorthandoff
364     \fi
365     \def\shorthandoff##1{%
366         \ifx\PackageWarning\undefined
367             \fb@warning{\noexpand\shorthandoff{;:!?} is helpless with
368                 LuaTeX,\ \ use \noexpand\NoAutoSpacing
369                 *inside a group* instead.}%
370         \else
371             \PackageWarning{frenchb.lfd}{\protect\shorthandoff{;:!?} is
372                 helpless with LuaTeX,\MessageBreak use \protect\NoAutoSpacing
373                 \space *inside a group* instead;\MessageBreak reported}%
374         \fi}%
375     \def\shorthandon##1{%
376     }
377 \newcommand*{\FB@luatex@punct@nonfrench}{%
378     \ifx\shorthandoffORI\undefined
379     \else
380         \let\shorthandon\shorthandonORI
381         \let\shorthandoff\shorthandoffORI
382     \fi
383 }
384 \FB@addto{extras}{\FB@luatex@punct@french}
385 \FB@addto{noextras}{\FB@luatex@punct@nonfrench}

```

In  $\LaTeX 2_{\epsilon}$ , file `frenchb.lua` will be loaded 'AtBeginDocument' *after* processing options (`ThinColonSpace` needs to be taken into account). The next definition will be used to activate Lua punctuation: it sets the language number for French, loads `frenchb.lua` and adds function `french_punctuation` at the end of the kerning

callback (no priority).

```
386 \def\activate@luatexpunct{%
387   \directlua{%
388     FR = \the\l@french
389     local path = kpse.find_file("frenchb.lua", "lua")
390     if path then
391       local f = dofile(path)
392       luatexbase.add_to_callback("kerning",
393         f, "frenchb.french_punctuation")
394     else
395       texio.write_nl('')
396       texio.write_nl('*****')
397       texio.write_nl('Error: frenchb.lua not found.')
398       texio.write_nl('*****')
399       texio.write_nl('')
400     end
401   }%
402 }
403 \fi
```

End of specific code for punctuation with LuaTeX engines.

## 2.2.2 Punctuation with XeTeX

If `\XeTeXinterchartokenstate` is available, we use the “inter char” mechanism to provide correct spacing in French before the four characters ; ! ? and :. The basis of the following code was borrowed from the `polyglossia` package, see `gloss-french.ldf`. 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 2.10).

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 ‘high 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.

The following part holds specific code for punctuation with XeTeX engines.

```
404 \newcount\FB@interchartokenstateORI
405 \ifFB@xetex@punct
406   \ifLaTeXe
407     \PackageInfo{frenchb.ldf}{No need for active punctuation characters%
408       \MessageBreak with this version of XeTeX!%
409       \MessageBreak reported}
410   \else
411     \fb@info{No need for active punctuation characters\
412       with this version of XeTeX!}
413   \fi
```

Six new character classes are defined for frenchb.



```

414 \newXeTeXintercharclass\FB@punctthick
415 \newXeTeXintercharclass\FB@punctthin
416 \newXeTeXintercharclass\FB@punctnul
417 \newXeTeXintercharclass\FB@guilo
418 \newXeTeXintercharclass\FB@guilf
419 \newXeTeXintercharclass\FB@guilnul

```

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.

```

420 \def\FB@charclassesORI{}
421 \def\empty{}
422 \def\FB@parse#1,#2\endparse{\def\FB@class{#1}%
423     \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.

```

424 \newcommand*\FB@xetex@punct@french){%

```

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

```

425 \ifx\FB@charclassesORI\empty
426     \FB@interchartokenstateORI=\XeTeXinterchartokenstate
427     \bbl@for\FB@char
428         {\':,\';,\!','\?', "AB,"BB,%
429             '\(,\'[,\'\{,\',,\'.,\'-,\'),'\],'\},%
430             '\%, "22,"27,"60,"2019,"A0,"202F}%
431         {\edef\FB@charclassesORI{\FB@charclassesORI%
432             \theXeTeXcharclass\FB@char,}}%
433     \let\shorthandonORI\shorthandon
434     \let\shorthandoffORI\shorthandoff
435 \fi

```

Set the classes and interactions between classes.

```

436 \XeTeXinterchartokenstate=1
437 \XeTeXcharclass '\: = \FB@punctthick
438 \XeTeXinterchartoks \z@ \FB@punctthick = {%
439     \ifhmode\FDP@colonspace\fi}%
440 \XeTeXinterchartoks \FB@guilf \FB@punctthick = {%
441     \FDP@colonspace}%
442 \XeTeXinterchartoks 255 \FB@punctthick = {%
443     \ifhmode\unskip\penalty\@M\FBcolonspace\fi}%
444 \bbl@for\FB@char
445     {\':,\';,\!','\?}%
446     {\XeTeXcharclass\FB@char=\FB@punctthin}%
447 \XeTeXinterchartoks \z@ \FB@punctthin = {%
448     \ifhmode\FDP@thinspace\fi}%
449 \XeTeXinterchartoks \FB@guilf \FB@punctthin = {%
450     \FDP@thinspace}%
451 \XeTeXinterchartoks 255 \FB@punctthin = {%
452     \ifhmode\unskip\penalty\@M\FBthinspace\fi}%
453 \XeTeXinterchartoks \FB@guilo \z@ = {%
454     \ifFBAutoSpaceGuill\FBguillspace\fi}%

```

```

455 \XeTeXinterchartoks \FB@guilo 255 = {%
456     \ifFBAutoSpaceGuill\FBguillspace\ignorespaces\fi}%
457 \XeTeXinterchartoks \z@ \FB@guilf = {%
458     \ifFBAutoSpaceGuill\FBguillspace\fi}%
459 \XeTeXinterchartoks \FB@punctthin \FB@guilf = {%
460     \ifFBAutoSpaceGuill\FBguillspace\fi}%
461 \XeTeXinterchartoks 255 \FB@guilf = {%
462     \ifFBAutoSpaceGuill\unskip\FBguillspace\fi}%

```

This will avoid spurious spaces in (!), [?] and with Unicode nobreakspaces (U+00A0, U+202F):

```

463 \bbl@for\FB@char
464     {'\[, '\(, "A0, "202F}%
465     {\XeTeXcharclass\FB@char=\FB@punctnul}%

```

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

```

466 \bbl@for\FB@char
467     {'\{, '\., '\., '\-, '\), '\], '\}, '\%, "22, "27, "60, "2019}%
468     {\XeTeXcharclass\FB@char=\z@}%

```

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

```

469 \def\shorthandoff##1{%
470     \ifx\PackageWarning\@undefined
471     \fb@warning{\noexpand\shorthandoff{;:!?} is helpless with
472     XeTeX,\@ use \noexpand\NoAutoSpacing
473     *inside a group* instead.}%
474     \else
475     \PackageWarning{frenchb.ldf}{\protect\shorthandoff{;:!?} is
476     helpless with XeTeX,\MessageBreak use \protect\NoAutoSpacing
477     \space *inside a group* instead;\MessageBreak reported}%
478     \fi}%
479 \def\shorthandon##1{%
480 }

```

`\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', the test on `\FB@charclassesORI` is mandatory.

```

481 \newcommand*\FB@xetex@punct@nonfrench{%
482     \ifx\FB@charclassesORI\empty
483     \else
484     \bbl@for\FB@char
485     {'\:, '\;, '\!, '\?, "AB, "BB, %
486     '\(, '\[, '\{, '\., '\., '\-, '\), '\], '\}, %
487     '\%, "22, "27, "60, "2019, "A0, "202F}%
488     {\expandafter\FB@parse\FB@charclassesORI\endparse
489     \XeTeXcharclass\FB@char=\FB@class}%
490     \def\FB@charclassesORI{%
491     \XeTeXinterchartokenstate=\FB@interchartokenstateORI
492     \let\shorthandon\shorthandonORI
493     \let\shorthandoff\shorthandoffORI

```

```

494     \fi
495   }
496   \FB@addto{extras}{\FB@xetex@punct@french}
497   \FB@addto{noextras}{\FB@xetex@punct@nonfrench}

```

End of specific code for punctuation with modern XeTeX engines.

```
498 \fi
```

### 2.2.3 Punctuation with standard (pdf)TeX

In standard (pdf)TeX we need to make the four characters ; ! ? and : ‘active’ and provide their definitions.

```

499 \ifFB@active@punct
500   \initiate@active@char{:}%
501   \initiate@active@char{;}%
502   \initiate@active@char{!}%
503   \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 `\FBthinspace` instead. If no space has been typed, we add `\FDP@thinspace` which will be defined, up to the user’s wishes, as `\FBthinspace`, or as `\@empty`.

```

504   \declare@shorthand{french}{;}{%
505     \ifhmode
506       \ifdim\lastskip>\z@
507         \unskip\penalty\@M\FBthinspace
508       \else
509         \FDP@thinspace
510     \fi
511   \fi

```

Now we can insert a ; character.

```
512   \string;}

```

The next three definitions are very similar.

```

513   \declare@shorthand{french}{!}%
514     \ifhmode
515       \ifdim\lastskip>\z@
516         \unskip\penalty\@M\FBthinspace
517       \else
518         \FDP@thinspace
519     \fi
520   \fi
521   \string!}
522   \declare@shorthand{french}{?}%
523     \ifhmode
524       \ifdim\lastskip>\z@
525         \unskip\penalty\@M\FBthinspace
526       \else

```

```

527         \FDP@thinspace
528         \fi
529     \fi
530     \string?}
531 \declare@shorthand{french}{:}{}%
532     \ifhmode
533         \ifdim\lastskip>\z@
534             \unskip\penalty\M\FBcolonspace
535         \else
536             \FDP@colonspace
537         \fi
538     \fi
539     \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.

```

540 \declare@shorthand{system}{:}{\string;}
541 \declare@shorthand{system}{!}{\string!}
542 \declare@shorthand{system}{?}{\string?}
543 \declare@shorthand{system}{;}{\string;}
544 %}

```

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

```

545 \FB@addto{extras}{\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.

```

546     \bbl@activate{:}\bbl@activate{;}%
547     \bbl@activate{!}\bbl@activate{?}%
548 }
549 \FB@addto{noextras}{%
550     \bbl@deactivate{:}\bbl@deactivate{;}%
551     \bbl@deactivate{!}\bbl@deactivate{?}%
552 }
553 \fi

```

#### 2.2.4 Punctuation switches common to all engines

A new ‘if’ `\ifFBAutoSpacePunctuation` needs to be defined now to control the two possible ways of dealing with ‘high punctuation’. its default value is true, but it can be set to false by `\frenchbsetup{AutoSpacePunctuation=false}` for finer control.

```

554 \newif\ifFBAutoSpacePunctuation \FBAutoSpacePunctuationtrue

```

`\AutoSpaceBeforeFDP` `\autospace@beforeFDP` and `\noautospace@beforeFDP` are internal commands. `\NoAutoSpaceBeforeFDP` `\autospace@beforeFDP` defines `\FDP@thinspace` and `\FDP@colonspace` as unbreakable spaces and sets LuaTeX attribute `\FB@addDPspace` to 1 (true), while `\noautospace@beforeFDP` lets these spaces empty and sets flag `\FB@addDPspace` to 0 (false). User commands `\AutoSpaceBeforeFDP` and `\NoAutoSpaceBeforeFDP`

do the same and take care of the flag `\ifFBAutoSpacePunctuation` in  $\LaTeX$ . Set the default now for Plain (done later for  $\LaTeX$ ).

```

555 \def\autospace@beforeFDP{%
556     \ifFB@luatex@punct\FB@addDPspace=1 \fi
557     \def\FDP@thinspace{\penalty\@M\FBthinspace}%
558     \def\FDP@colonspace{\penalty\@M\FBcolonspace}}
559 \def\noautospace@beforeFDP{%
560     \ifFB@luatex@punct\FB@addDPspace=0 \fi
561     \let\FDP@thinspace\@empty
562     \let\FDP@colonspace\@empty}
563 \ifLaTeXe
564     \def\AutoSpaceBeforeFDP{\autospace@beforeFDP
565                             \FBAutoSpacePunctuationtrue}
566     \def\NoAutoSpaceBeforeFDP{\noautospace@beforeFDP
567                               \FBAutoSpacePunctuationfalse}
568     \AtEndOfPackage{\AutoSpaceBeforeFDP}
569 \else
570     \let\AutoSpaceBeforeFDP\autospace@beforeFDP
571     \let\NoAutoSpaceBeforeFDP\noautospace@beforeFDP
572     \AutoSpaceBeforeFDP
573 \fi

```

In  $\LaTeX_{2\epsilon}$  `\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.

```

574 \ifLaTeXe
575     \DeclareRobustCommand\ttfamilyFB{%
576         \FBAutoSpaceGuillfalse
577         \ifFB@luatex@punct\FB@addGUILspace=0 \fi
578         \noautospace@beforeFDP\ttfamilyORI}%
579     \DeclareRobustCommand\rmfamilyFB{%
580         \FBAutoSpaceGuilltrue
581         \ifFB@luatex@punct\FB@addGUILspace=1 \fi
582         \ifFBAutoSpacePunctuation
583             \autospace@beforeFDP
584         \else
585             \noautospace@beforeFDP
586         \fi
587         \rmfamilyORI}%
588     \DeclareRobustCommand\sffamilyFB{%
589         \FBAutoSpaceGuilltrue
590         \ifFB@luatex@punct\FB@addGUILspace=1 \fi

```

```

591         \ifFBAutoSpacePunctuation
592         \autospace@beforeFDP
593         \else
594         \noautospace@beforeFDP
595         \fi
596         \sffamilyORI}%
597 \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 TeX, LuaTeX and XeTeX based engines) and is meant to be used inside a group.

```

598 \newcommand*{\NoAutoSpacing}{\FBAutoSpaceGuillfalse
599   \ifFB@active@punct\shorthandoff{;:!?}\fi
600   \ifFB@xetex@punct\XeTeXinterchartokenstate=0 \fi
601   \ifFB@luatex@punct\FB@addDPspace=0 \FB@addGUILspace=0 \fi
602 }

```

### 2.3 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 French is to use the command **\frquote** (see below). Dummy definition of **\og** and **\fg** just to ensure that this commands are not yet defined. The default definition of **\og** and **\fg** will be set later (for English) by **\bbl@nonfrenchguillemets**.

```

603 \newcommand*{\og}{\@empty}
604 \newcommand*{\fg}{\@empty}

```

**\guillemotleft** **\guillemotright** **\textquotedblleft** **\textquotedblright** **LaTeX** users are supposed to use 8-bit output encodings (T1, LY1, ...) to typeset French, those who still stick to OT1 should call **aeguill** or a similar package. In both cases the commands **\guillemotleft** and **\guillemotright** will print the French opening and closing quote characters from the output font. For XeLaTeX and LuaLaTeX, **\guillemotleft** and **\guillemotright** are defined by package **xunicode** loaded by **fontspec**.

We provide the following definitions for non-LaTeX users only as fall-back, they are welcome to change them for anything better.

```

605 \ifLaTeXe
606 \else
607   \ifFBunicode
608     \def\guillemotleft{{\char"00AB}}
609     \def\guillemotright{{\char"00BB}}
610     \def\textquotedblleft{{\char"201C}}
611     \def\textquotedblright{{\char"201D}}
612   \else
613     \def\guillemotleft{\leavevmode\raise0.25ex
614       \hbox{\scriptscriptstyle\ll}}
615     \def\guillemotright{\raise0.25ex
616       \hbox{\scriptscriptstyle\gg}}

```

```

617 \def\textquotedblleft{''}
618 \def\textquotedblright{''}
619 \fi
620 \let\xspace\relax
621 \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. `\FBguillspace` which does the spacing, has been fine tuned by Thierry Bouche to 80% of an inter-word space but with reduced stretchability. 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).

LuaTeX which requires skips; `\FBguillskip` is computed from `\FBguillspace` for the `lmr10` font, its dimensions will be scaled by `frenchb.lua` for the current font.

```

622 \newskip\FBguillskip
623 \FBguillskip=2.664pt plus 0.500pt minus 0.888pt \relax
624 \newcommand*{\FBguillspace}{\penalty\@M\hskip.8\fontdimen2\font
625                                     plus.3\fontdimen3\font
626                                     minus.8\fontdimen4\font}

```

`\FBguillspace` is not used with LuaTeX.

```

627 \ifFB@luatex@punct
628 \DeclareRobustCommand*{\FB@og}{\leavevmode
629     \bgroup\FB@addGUllspace=1 \guillemotleft\egroup}
630 \DeclareRobustCommand*{\FB@fg}{\ifdim\lastskip>\z@\unskip\fi
631     \bgroup\FB@addGUllspace=1 \guillemotright\egroup\xspace}
632 \fi

```

With XeTeX, `\FBAutoSpaceGuill` is set to `false` locally to prevent the quotes characters from adding space when option `og=«`, `fg=»` is set. characters.

```

633 \ifFB@xetex@punct
634 \DeclareRobustCommand*{\FB@og}{\leavevmode
635     \bgroup\FBAutoSpaceGuillfalse\guillemotleft\egroup
636     \FBguillspace}
637 \DeclareRobustCommand*{\FB@fg}{\ifdim\lastskip>\z@\unskip\fi
638     \FBguillspace
639     \bgroup\FBAutoSpaceGuillfalse\guillemotright\egroup\xspace}
640 \fi
641 \ifFB@active@punct
642 \DeclareRobustCommand*{\FB@og}{\leavevmode
643     \guillemotleft
644     \FBguillspace}
645 \DeclareRobustCommand*{\FB@fg}{\ifdim\lastskip>\z@\unskip\fi
646     \FBguillspace
647     \guillemotright\xspace}
648 \fi

```

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

```

649 \ifLaTeXe
650 \def\bbf@frenchguillemets{\renewcommand*\og{\FB@og}%
651 \renewcommand*\fg{\FB@fg}}
652 \renewcommand*\og{\textquotedblleft}
653 \renewcommand*\fg{\ifdim\lastskip>\z@\unskip\fi \textquotedblright}
654 \else
655 \def\bbf@frenchguillemets{\let\og\FB@og
656 \let\fg\FB@fg}
657 \def\og{\textquotedblleft}
658 \def\fg{\ifdim\lastskip>\z@\unskip\fi\textquotedblright}
659 \fi
660 \FB@addto{extras}{\babel@save\og \babel@save\fg \bbf@frenchguillemets}

```

`\frquote` Maximum two levels are supported by `\frquote{}`. Let's define the default quote characters to be used for level one or two of quotes. . .

```

661 \newcommand*\ogi{\FB@og}
662 \newcommand*\fgi{\FB@fg}
663 \newcommand*\ogii{\textquotedblleft}
664 \newcommand*\fgii{\textquotedblright}

```

and the needed technical stuff to handle options:

```

665 \newcount\FBguill@level
666 \newif\ifFBcloseguill \FBcloseguilltrue
667 \newif\ifFBInnerGuillSingle
668 \def\FBguillopen{\bgroup\NoAutoSpacing\guillemotleft\egroup}
669 \def\FBguillclose{\bgroup\NoAutoSpacing\guillemotright\egroup}
670 \let\FBguillnone\relax
671 \let\FBeveryparguill\FBguillopen
672 \ifFB@luatex@punct
673 \let\FBverylineguill\FBguillopen
674 \else
675 \let\FBverylineguill\FBguillnone
676 \fi

```

The main command `\frquote` accepts (in  $\text{\LaTeX}2_{\epsilon}$  only) a starred version which suppresses the closing quote; it is meant to be used for inner quotations which end together with the outer one, then only one closing guillemet (the outer one) should be printed.

```

677 \ifLaTeXe
678 \DeclareRobustCommand\frquote{%
679 \@ifstar{\FBcloseguillfalse\fr@quote}%
680 {\FBcloseguilltrue\fr@quote}}
681 \else
682 \newcommand\frquote[1]{\fr@quote{#1}}
683 \fi

```

The internal command `\fr@quote` takes one (long) argument: the quotation text.

```

684 \newcommand{\fr@quote}[1]{%

```



```

685 \bgroup
686 \ifnum\FBguill@level=2
687 \PackageWarning{frenchb.ldf}{%
688 \protect\frquote\space accepts no more than two levels
689 \MessageBreak of quotations. Reported}
690 \else
691 \advance\FBguill@level by \@ne
692 \fi
693 \ifnum\FBguill@level=1

```

Set \FBeverypar@quote for level 1 quotations:

```

694 \ifx\FBeveryparguill\relax
695 \let\FBeverypar@quote\relax
696 \else
697 \def\FBeverypar@quote{\FBeveryparguill
698 \kern.8\fontdimen2\font}%
699 \fi
700 \logi
701 \everypar \expandafter{\the\everypar \FBeverypar@quote}%
702 #1\fgi
703 \else

```

This for level 2 (inner) quotations: Omega's command \localleftbox (included in LuaTeX, renamed \luatexlocalleftbox in LuaLaTeX) is convenient for repeating guillemets at the beginning of every line.

```

704 \ifx\FBverylineguill\FBguillopen
705 \luatexlocalleftbox{\guillemotleft\kern.8\fontdimen2\font}%
706 \let\FBeverypar@quote\relax
707 \FB@og #1\ifFBcloseguill\FB@fg\fi
708 \else
709 \ifx\FBverylineguill\FBguillclose
710 \luatexlocalleftbox{\guillemotright\kern.8\fontdimen2\font}%
711 \let\FBeverypar@quote\relax
712 \FB@og #1\ifFBcloseguill\FB@fg\fi
713 \else

```

otherwise we need to redefine \FBeverypar@quote (and eventually \logii, \fgii) for level 2 quotations:

```

714 \let\FBeverypar@quote\relax
715 \ifFBInnerGuillSingle
716 \def\logii{\leavevmode
717 \guilsinglleft\FBguillspace}%
718 \def\fgii{\ifdim\lastskip>\z@\unskip\fi
719 \FBguillspace\guilsinglright}%
720 \ifx\FBeveryparguill\FBguillopen
721 \def\FBeverypar@quote{\guilsinglleft
722 \kern.8\fontdimen2\font}%
723 \fi
724 \ifx\FBeveryparguill\FBguillclose
725 \def\FBeverypar@quote{\guilsinglright
726 \kern.8\fontdimen2\font}%
727 \fi

```

```

728         \fi
729         \ogii #1\ifFBcloseguill \fgii \fi
730     \fi
731 \fi
732 \fi
733 \egroup
734 }

```

## 2.4 Date in French

`\datefrench` The macro `\datefrench` redefines the command `\today` to produce French dates. This new implementation requires babel 3.9i or newer but, as of 3.9k, doesn't work with Plain based formats, so `\date\CurrentOption` is defined the old way for these formats.

```

735 \ifLaTeXe
736 \def\BabelLanguages{french,acadian}
737 \StartBabelCommands*{\BabelLanguages}{date}
738     [unicode, fontenc=EU1 EU2, charset=utf8]
739     \SetString\monthiiname{février}
740     \SetString\monthviiname{août}
741     \SetString\monthxiiname{décembre}
742 \StartBabelCommands*{\BabelLanguages}{date}
743     \SetStringLoop{month#1name}{%
744         janvier,f\'evrier,mars,avril,mai,juin,juillet,%
745         ao^ut,septembre,octobre,novembre,d\'ecembre}
746     \SetString\today{{\number\day}\ifnum1=\day {\ier}\fi \space
747         \csname month\romannumeral\month name\endcsname \space
748         \number\year
749     }
750 \EndBabelCommands
751 \else
752 \ifFBunicode
753     \@namedef{date\CurrentOption}{%
754         \def\today{{\number\day}\ifnum1=\day {\ier}\fi \space
755             \ifcase\month
756                 \or janvier\or février\or mars\or avril\or mai\or
757                 juin\or juillet\or août\or septembre\or
758                 octobre\or novembre\or décembre\fi
759             \space \number\year}}
760 \else
761     \@namedef{date\CurrentOption}{%
762         \def\today{{\number\day}\ifnum1=\day {\ier}\fi \space
763             \ifcase\month
764                 \or janvier\or f\'evrier\or mars\or avril\or mai\or
765                 juin\or juillet\or ao^ut\or septembre\or
766                 octobre\or novembre\or d\'ecembre\fi
767             \space \number\year}}
768 \fi
769 \fi

```

## 2.5 Extra utilities

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

`\up` eases the typesetting of superscripts like '1<sup>er</sup>'. Up to version 2.0 of frenchb `\up` was just a shortcut for `\textsuperscript` in L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>, 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).

```
770 \newif\ifFB@poorman
771 \newdimen\FB@Mht
772 \ifLaTeXe
773 \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{}`.

```
774 \newcommand*{\FBsupR}{-0.12}
775 \newcommand*{\FBsupS}{0.65}
776 \newcommand*{\FB@lc}[1]{\MakeLowercase{#1}}
777 \DeclareRobustCommand*{\FB@up@fake}[1]{%
778   \settoheight{\FB@Mht}{M}%
779   \addtolength{\FB@Mht}{\FBsupR \FB@Mht}%
780   \addtolength{\FB@Mht}{-\FBsupS ex}%
781   \raisebox{\FB@Mht}{\scalefont{\FBsupS}{\FB@lc{#1}}}%
782 }
```

The only packages I currently know to take advantage of real superscripts are a) `realscripts` used in conjunction with XeLaTeX or LuaLaTeX and OpenType fonts having the font feature 'VerticalPosition=Superior' 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.

```

783 \def\FB@split#1#2#3#4\@nil{\def\FB@firstthree{#1#2#3}%
784 \def\FB@suffix{#4}}
785 \def\FB@x{x}
786 \def\FB@j{j}
787 \DeclareRobustCommand*\FB@up[1]{%
788 \bgroup \FB@poormantrue
789 \expandafter\FB@split\fb@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.

```

790 \edef\reserved@a{\lowercase{%
791 \noexpand\IfFileExists{\fb@encoding\FB@firstthree -sup.fd}}}%
792 \reserved@a
793 {\ifx\FB@suffix\FB@x \FB@poormanfalse\fi
794 \ifx\FB@suffix\FB@j \FB@poormanfalse\fi
795 \ifFB@poorman \FB@up@fake{#1}%
796 \else \FB@up@real{#1}%
797 \fi}%
798 {\FB@up@fake{#1}}%
799 \egroup}

```

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

```

800 \newcommand*\FB@up@real[1]{\bgroup
801 \fontfamily{\FB@firstthree -sup}\selectfont \FB@lc{#1}\egroup}

```

`\fup` is defined as `\FB@up` unless `\realsuperscript` is defined by `realscripts.sty`.

```

802 \DeclareRobustCommand*\fup[1]{%
803 \ifx\realsuperscript\@undefined
804 \FB@up{#1}%
805 \else
806 \bgroup\let\fakesuperscript\FB@up@fake
807 \realsuperscript{\FB@lc{#1}}\egroup
808 \fi}

```

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

```

809 \providecommand*\up{\relax}

```

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

```

810 \else
811 \providecommand*\up[1]{\leavevmode\raise1ex\hbox{\sevenrm #1}}
812 \fi

```

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

```

\ier 813 \def\ieme{\up{e}\xspace}
\iere 814 \def\ienes{\up{es}\xspace}
\iemes 815 \def\ier{\up{er}\xspace}
\iers 816 \def\iers{\up{ers}\xspace}
\ieres

```

```
817 \def\iere{\up{re}\xspace}
818 \def\ieres{\up{res}\xspace}
```

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

```
\no 819 \newcommand*\FrenchEnumerate}[1]{%
\nos 820      #1\up{o}\kern+.3em}
\nos 821 \newcommand*\FrenchPopularEnumerate}[1]{%
\primo 822      #1\up{o})\kern+.3em}
```

**\fprimo**) Typing \primo should result in ‘1°’,

```
823 \def\primo{\FrenchEnumerate1}
824 \def\secundo{\FrenchEnumerate2}
825 \def\tertio{\FrenchEnumerate3}
826 \def\quarto{\FrenchEnumerate4}
```

while typing \fprimo) gives ‘1°’.

```
827 \def\fprimo{\FrenchPopularEnumerate1}
828 \def\fsecundo{\FrenchPopularEnumerate2}
829 \def\ftertio{\FrenchPopularEnumerate3}
830 \def\fquarto{\FrenchPopularEnumerate4}
```

Let’s provide four macros for the common abbreviations of “Numéro”.

```
831 \DeclareRobustCommand*\No{\N\up{o}\kern+.2em}
832 \DeclareRobustCommand*\no{\n\up{o}\kern+.2em}
833 \DeclareRobustCommand*\Nos{\N\up{os}\kern+.2em}
834 \DeclareRobustCommand*\nos{\n\up{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}.

```
835 \DeclareRobustCommand*\bsc}[1]{\leavevmode\beginngroup\kern0pt
836      \scshape #1\endgroup}
837 \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.

```
838 \ifBUnicode
839   \newcommand*\at{{\char"0040}}
840   \newcommand*\circonflexe{{\char"005E}}
841   \newcommand*\tild{{\char"007E}}
842   \newcommand*\boi{\textbackslash}
843   \newcommand*\degree{{\char"00B0}}
844 \else
845   \ifLaTeXe
846     \DeclareTextSymbol{\at}{T1}{64}
```

```

847 \DeclareTextSymbol{\circflxe}{T1}{94}
848 \DeclareTextSymbol{\tilde}{T1}{126}
849 \DeclareTextSymbolDefault{\at}{T1}
850 \DeclareTextSymbolDefault{\circflxe}{T1}
851 \DeclareTextSymbolDefault{\tilde}{T1}
852 \DeclareRobustCommand*\boi{\textbackslash}
853 \DeclareRobustCommand*\degre}{\r{}}
854 \else
855 \def\T@one{T1}
856 \ifx\fontencoding\T@one
857 \newcommand*\degre}{{\char6}}
858 \else
859 \newcommand*\degre}{{\char23}}
860 \fi
861 \newcommand*\at}{{\char64}}
862 \newcommand*\circflxe}{{\char94}}
863 \newcommand*\tilde}{{\char126}}
864 \newcommand*\boi}{\backslash}
865 \fi
866 \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 0.3 em, 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 emulating ‘degrees’ from the `\r{}` accent. Otherwise we advise the user (once only) to use TS1-encoding.

```

867 \ifLaTeXe
868 \newcommand*\degrees}{\degre}
869 \ifFBunicode
870 \DeclareRobustCommand*\degrees}{\degre}
871 \else
872 \def\Warning@degree@TSone{%
873 \PackageWarning{frenchb.ldf}{%
874 Degrees would look better in TS1-encoding:%
875 \MessageBreak add \protect
876 \usepackage{textcomp} to the preamble.%
877 \MessageBreak Degrees used}}
878 \AtBeginDocument{\ifx\DeclareEncodingSubset\undefined
879 \DeclareRobustCommand*\degrees}{%
880 \leavevmode\hbox to 0.3em{\hss\degre\hss}%
881 \Warning@degree@TSone
882 \global\let\Warning@degree@TSone\relax}%
883 \else
884 \DeclareRobustCommand*\degrees}{%
885 \hbox{\UseTextSymbol{TS1}{\textdegree}}}%
886 \fi
887 }

```

```

888 \fi
889 \else
890 \newcommand*{\degrees}{%
891 \leavevmode\hbox to 0.3em{\hss\degre\hss}}
892 \fi

```

## 2.6 Formatting numbers

`\DecimalMathComma` As mentioned in the T<sub>E</sub>Xbook p. 134, the comma is of type `\mathpunct` in math mode: `\StandardMathComma` 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.

```

893 \newcount\std@mcc
894 \newcount\dec@mcc
895 \std@mcc=\mathcode'\,
896 \dec@mcc=\std@mcc
897 \@tempcnta=\std@mcc
898 \divide\@tempcnta by "1000
899 \multiply\@tempcnta by "1000
900 \advance\dec@mcc by -\@tempcnta
901 \newcommand*{\DecimalMathComma}{\iflanguage{french}%
902 \FB@addto{extras}{\mathcode'\,=\dec@mcc}}%
903 \FB@addto{extras}{\mathcode'\,=\dec@mcc}%
904 }
905 \newcommand*{\StandardMathComma}{\mathcode'\,=\std@mcc
906 \FB@addto{extras}{\mathcode'\,=\std@mcc}%
907 }
908 \FB@addto{noextras}{\mathcode'\,=\std@mcc}

```

`\nombre` The command `\nombre` is now borrowed from `numprint.sty` for L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>. 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.

```

909 \newcommand*{\nombre}[1]{\fb@warning{*** \noexpand\nombre
910 no longer formats numbers\string! ***}}

```

The next definitions only make sense for L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>. For Plain based formats, let's activate LuaTeX punctuation if necessary, then cleanup and exit. Temporary fix: `\l@french` is not properly set by babel 3.9h with Plain LuaTeX format.

```

911 \let\FBstop@here\relax
912 \def\FBclean@on@exit{\let\ifLaTeXe\undefined
913 \let\LaTeXettrue\undefined
914 \let\LaTeXefalse\undefined}
915 \ifx\magnification\@undefined
916 \else

```

```

917 \def\FBstop@here{\ifFB@luatex@punct
918             \activate@luatexpunct
919             \fi
920             \FBClean@on@exit
921             \ldf@quit\CurrentOption\endinput}
922 \fi
923 \FBstop@here

```

What follows is for  $\LaTeX 2_\epsilon$  *only*; as all  $\LaTeX 2_\epsilon$  based formats include  $\epsilon\text{-TeX}$ , we can use `\ifdefined` now. We redefine `\nombre` for  $\LaTeX 2_\epsilon$ . 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.

```

924 \renewcommand*{\nombre}[1]{\Warning@nombre\numprint{#1}}
925 \newcommand*{\Warning@nombre}{%
926   \ifdefined\numprint
927   \else
928     \PackageWarning{frenchb.ldf}{%
929       \protect\nombre\space now relies on package numprint.sty,%
930       \MessageBreak add \protect
931       \usepackage[autolanguage]{numprint}\MessageBreak
932       to your preamble *after* loading babel,\MessageBreak
933       see file numprint.pdf for more options.\MessageBreak
934       \protect\nombre\space called}%
935   \global\let\Warning@nombre\relax
936   \fi
937 }

```

## 2.7 Caption names

The next step consists in defining the French equivalents for the  $\LaTeX$  caption names.

`\captionsfrench` Let's first define `\captionsfrench` which sets all strings used in the four standard document classes provided with  $\LaTeX$ .

Let's give a chance to a class or a package read before `frenchb` to define `\FBfigtabshape` as `\relax`, otherwise `\FBfigtabshape` will be defined as `\scshape` (can be changed with `\frenchbsetup{SmallCapsFigTabCaptions=false}`).

```

938 \ifx\FBfigtabshape@undefined \let\FBfigtabshape\scshape \fi

```

New implementation for caption names (requires `babel`'s 3.9 or up).

```

939 \StartBabelCommands*{\BabelLanguages}{captions}
940   [unicode, fontenc=EU1 EU2, charset=utf8]
941   \SetString{\refname}{Références}
942   \SetString{\abstractname}{Résumé}
943   \SetString{\prefacename}{Préface}
944   \SetString{\contentsname}{Table des matières}
945   \SetString{\ccname}{Copie à }
946   \SetString{\proofname}{Démonstration}
947   \SetStringLoop{ordinal#1}{%
948     Première,Deuxième,Troisième,Quatrième,Cinquième,%
949     Sixième,Septième,Huitième,Neuvième,Dixième,Onzième,%

```



```

950     Douzième,Treizième,Quatorzième,Quinzième,Seizième,%
951     Dix-septième,Dix-huitième,Dix-neuvième,Vingtième}
952 \StartBabelCommands*{\BabelLanguages}{captions}
953   \SetString{\refname}{R\`ef\`erences}
954   \SetString{\abstractname}{R\`esum\`e}
955   \SetString{\bibname}{Bibliographie}
956   \SetString{\prefacename}{Pr\`eface}
957   \SetString{\chaptername}{Chapitre}
958   \SetString{\appendixname}{Annexe}
959   \SetString{\contentsname}{Table des mati\`eres}
960   \SetString{\listfigurename}{Table des figures}
961   \SetString{\listtablename}{Liste des tableaux}
962   \SetString{\indexname}{Index}
963   \SetString{\figurename}{{\FBfigtabshape Figure}}
964   \SetString{\tablename}{{\FBfigtabshape Table}}
965   \SetString{\pagename}{page}
966   \SetString{\seename}{voir}
967   \SetString{\alsoname}{voir aussi}
968   \SetString{\enclname}{P.~J. }
969   \SetString{\ccname}{Copie `a }
970   \SetString{\headtoname}{}
971   \SetString{\proofname}{D\`emonstration}
972   \SetString{\glossaryname}{Glossaire}

```

When `PartNameFull=true` (default), `\part{}` is printed in French as “Première partie” instead of “Partie I”. As logic is prohibited inside `\SetString`, let’s hide the test about `PartNameFull` in `\FB@partname`.

```

973   \SetStringLoop{ordinal#1}{%
974     Premi\`ere,Deuxi\`eme,Troisi\`eme,Quatri\`eme,Cinqui\`eme,%
975     Sixi\`eme,Septi\`eme,Huiti\`eme,Neuvi\`eme,Dixi\`eme,Onzi\`eme,%
976     Douzi\`eme,Treizi\`eme,Quatorzi\`eme,Quinzi\`eme,Seizi\`eme,%
977     Dix-septi\`eme,Dix-huiti\`eme,Dix-neuvi\`eme,Vingti\`eme}
978 \AfterBabelCommands{%
979   \DeclareRobustCommand*{\FB@emptypart}{\def\thepart{}}%
980   \DeclareRobustCommand*{\FB@partname}{%
981     \ifFBPartNameFull
982       \csname ordinal\romannumeral\value{part}\endcsname\space
983       partie\FB@emptypart
984     \else
985       Partie%
986     \fi}%
987   }
988   \SetString{\partname}{\FB@partname}
989 \EndBabelCommands

```

The following patch is for koma-script classes: `\partformat` needs to be redefined in French as this command, defined as `\partname~\thepart\autodot` is incompatible with our redefinition of `\partname`. The code is postponed to the end of package because `\ifFB@koma` will be defined and set later on (see p. 43).

```

990 \AtEndOfPackage{%
991   \ifFB@koma

```

```

992     \ifdefined\partformat
993         \FB@addto{captions}{%
994             \ifFBPartNameFull
995                 \babel@save\partformat
996                 \renewcommand*{\partformat}{\partname}%
997             \fi}%
998     \fi
999 \fi
1000 }

```

Up to v2.6h frenchb used to merge `\captionsfrenchb` and `\captionspancais` into `\captionsfrench` at `\begin{document}`. This is deprecated in favor of the new (much simpler!) syntax introduced in babel 3.9. No need to define `\captionscanadien` and `\captionssacadian` either.

**\CaptionSeparator** Let's consider now captions in figures and tables. In French, captions in figures and tables should never be printed as 'Figure 1:' which is the default in standard  $\LaTeX 2_{\epsilon}$  classes; the ':' is made active too late, no space is added before it. With LuaLaTeX and XeLaTeX, this glitch doesn't occur, you get 'Figure 1 :' which is correct in French. With pdfLaTeX frenchb provides the following workaround.

The standard definition of `\@makecaption` (e.g., the one provided in `article.cls`, `report.cls`, `book.cls` which is frozen for  $\LaTeX 2_{\epsilon}$  according to Frank Mittelbach), is saved in `\STD@makecaption`. 'AtBeginDocument' we compare it to its current definition (some classes like `memoir`, `koma-script` classes, `AMS` classes, `ua-thesis.cls`... change it). If they are identical, frenchb just adds a hook called `\FBCaption@Separator` to `\@makecaption`; `\FBCaption@Separator` defaults to ':' as in the standard `\@makecaption` and will be changed to ':' in French 'AtBeginDocument'; it can be also set to `\CaptionSeparator` ('-') using [CustomiseFigTabCaptions](#). While saving the standard definition of `\@makecaption` we have to make sure that characters ':' and '>' have `\catcode 12` (frenchb makes ':' active and `spanish.ldf` makes '>' active).

```

1001 \bgroup
1002 \catcode':=12 \catcode'>=12 \relax
1003 \long\gdef\STD@makecaption#1#2{%
1004     \vskip\abovecaptionskip
1005     \sbox\@tempboxa{#1: #2}%
1006     \ifdim \wd\@tempboxa >\hspace
1007         #1: #2\par
1008     \else
1009         \global \@minipagefalse
1010         \hb@xt@\hspace{\hfil\box\@tempboxa\hfil}%
1011     \fi
1012     \vskip\belowcaptionskip}
1013 \egroup

```

The `caption` and `floatrow` packages are compatible with frenchb if they are loaded after babel (a warning is printed in the .log file when they are loaded too early).

No warning is issued for `SMF` and `AMS` classes as their layout of captions is compatible with French typographic standards.

With memoir and koma-script classes, frenchb customises `\captiondelim` or `\captionformat` in French (unless option `CustomiseFigTabCaptions` is set to `false`) and issues no warning.

When `\@makecaption` has been changed by another class or package, a warning is printed in the .log file.

```

1014 \newif\if@FBwarning@capsep
1015 \@FBwarning@capseptrue
1016 \newcommand{\FBWarning}[2]{\PackageWarning{#1}{#2}}
1017 \newcommand*{\CaptionSeparator}{\space\textendash\space}
1018 \def\FBCaption@Separator{: }
1019 \long\def\FB@makecaption#1#2{%
1020   \vskip\abovecaptionskip
1021   \sbox\@tempboxa{#1\FBCaption@Separator #2}%
1022   \ifdim \wd\@tempboxa >\hsize
1023     #1\FBCaption@Separator #2\par
1024   \else
1025     \global \@minipagefalse
1026     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
1027   \fi
1028   \vskip\belowcaptionskip}

```

Disable the standard warning with AMS and SMF classes.

```

1029 \@ifclassloaded{amsart}{\@FBwarning@capsepfalse}{}
1030 \@ifclassloaded{amsbook}{\@FBwarning@capsepfalse}{}
1031 \@ifclassloaded{amsdtx}{\@FBwarning@capsepfalse}{}
1032 \@ifclassloaded{amslatex}{\@FBwarning@capsepfalse}{}
1033 \@ifclassloaded{amproc}{\@FBwarning@capsepfalse}{}
1034 \@ifclassloaded{smfart}{\@FBwarning@capsepfalse}{}
1035 \@ifclassloaded{smfbook}{\@FBwarning@capsepfalse}{}

```

Disable the standard warning unless high punctuation is active.

```

1036 \ifFB@active@punct\else\@FBwarning@capsepfalse\fi

```

No warning with memoir or koma-script classes: they change `\@makecaption` but we will manage to customise them in French later on (see below after executing `\FBprocess@options`).

```

1037 \newif\ifFB@koma
1038 \@ifclassloaded{memoir}{\@FBwarning@capsepfalse}{}
1039 \@ifclassloaded{scrartcl}{\@FBwarning@capsepfalse\FB@komatrue}{}
1040 \@ifclassloaded{scrbook}{\@FBwarning@capsepfalse\FB@komatrue}{}
1041 \@ifclassloaded{scrreprt}{\@FBwarning@capsepfalse\FB@komatrue}{}

```

No warning with the beamer class which defines `\beamer@makecaption` (customised below) instead of `\@makecaption`. No warning either if `\@makecaption` is undefined (i.e. letter).

```

1042 \@ifclassloaded{beamer}{\@FBwarning@capsepfalse}{}
1043 \ifdefined\@makecaption\else\@FBwarning@capsepfalse\fi

```

Check if package caption is loaded now (before babel/frenchb), then issue a warning advising to load it after babel/frenchb and disable the standard warning.

```

1044 \@ifpackageloaded{caption}

```

```

1045   {\FBWarning{frenchb.ldf}%
1046     {Please load the "caption" package\MessageBreak
1047       AFTER babel/frenchb; reported}%
1048   \@FBwarning@capsepfalse}%
1049   {}}

```

Same for package floatrow.

```

1050 \@ifpackageloaded{floatrow}
1051   {\FBWarning{frenchb.ldf}%
1052     {Please load the "floatrow" package\MessageBreak
1053       AFTER babel/frenchb; reported}%
1054   \@FBwarning@capsepfalse}%
1055   {}}

```

First check the definition of \@makecaption, change it or issue a warning in case it has been changed by a class or package not (yet) compatible with frenchb; then change the definition of \FBCaption@Separator, taking care that the colon is typeset correctly in French (*not* ‘Figure 1: légende’).

```

1056 \@AtBeginDocument{%
1057   \ifx\@makecaption\STD@makecaption
1058     \global\let\@makecaption\FB@makecaption

```

Do not overwrite \FBCaption@Separator if already saved as ‘:’ for other languages and set to \CaptionSeparator by \extrasfrench when French is the main language.

```

1059   \ifFBoldFigTabCaptions
1060   \else
1061     \def\FBCaption@Separator{{\autospace@beforeFDP : }}%
1062   \fi
1063   \ifFBCustomiseFigTabCaptions
1064     \ifx\bbbl@main@language\FB@french
1065       \def\FBCaption@Separator{\CaptionSeparator}%
1066     \fi
1067   \fi
1068   \@FBwarning@capsepfalse
1069 \fi
1070 \if@FBwarning@capsep
1071   \FBWarning{frenchb.ldf}%
1072   {Figures' and tables' captions might look like\MessageBreak
1073     'Figure 1:' which is wrong in French.\MessageBreak
1074     Check your class or packages to change this;\MessageBreak
1075     reported}%
1076 \fi
1077 \let\FB@makecaption\relax
1078 \let\STD@makecaption\relax
1079 }

```

## 2.8 Dots...

`\FBtextellipsis` L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>'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  $\LaTeX 2_\epsilon$  only).

The `\if` construction in the  $\LaTeX 2_\epsilon$  definition of `\dots` doesn't allow the use of `xspace` (`xspace` is always followed by a `\fi`), so we use the AMS- $\LaTeX$  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.

```
1080 \ifFBunicode
1081   \let\FBtextellipsis\textellipsis
1082 \else
1083   \DeclareTextSymbol{\FBtextellipsis}{LY1}{133}
1084   \DeclareTextCommandDefault{\FBtextellipsis}{%
1085     .\kern\fontdimen3\font.\kern\fontdimen3\font.\xspace}
1086 \fi
```

`\Mdots@` and `\Tdots@` hold the definitions of `\dots` in Math and Text mode. They default to those of `amsmath-2.0`, and will revert to standard  $\LaTeX$  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.

```
1087 \newcommand*\Tdots@{\@xp\textellipsis}
1088 \newcommand*\Mdots@{\@xp\mdots@}
1089 \AtBeginDocument{\DeclareRobustCommand*\dots}{\relax
1090                   \csname\ifmmode M\else T\fi dots@\endcsname}%
1091                   \ifdefined\@xp\else\let\@xp\relax\fi
1092                   \ifdefined\mdots@\else\let\Mdots@\mathellipsis\fi
1093                   }
1094 \def\bbbl@frenchdots{\babel@save\Tdots@ \let\Tdots@\FBtextellipsis}
1095 \FB@addto{extras}{\bbbl@frenchdots}
```

## 2.9 More checks about packages' loading order

Like packages `captions` and `floatrow` (see section 2.7), package listings should be loaded after `babel/frenchb` due to active characters issues (pdfLaTeX only).

```
1096 \ifFB@active@punct
1097   \@ifpackageloaded{listings}
1098     {\FBWarning{frenchb.ldf}%
1099       {Please load the "listings" package\MessageBreak
1100        AFTER babel/frenchb; reported}%
1101     }{}
1102 \fi
```

Package `natbib` should be loaded before `babel/frenchb` due to active characters issues (pdfLaTeX only).

```
1103 \newif\if@FBwarning@natbib
1104 \ifFB@active@punct
1105   \@ifpackageloaded{natbib}{}{\@FBwarning@natbibtrue}
1106 \fi
```

```

1107 \AtBeginDocument{%
1108   \if@FBwarning@natbib
1109     \@ifpackageloaded{natbib}{\@FBwarning@natbibfalse}%
1110   \fi
1111   \if@FBwarning@natbib
1112     \FBWarning{frenchb.ldf}%
1113     {Please load the "natbib" package\MessageBreak
1114      BEFORE babel/frenchb; reported}%
1115   \fi
1116 }

```

Package numprint with option autolanguage should be loaded after babel/frenchb (all engines).

```

1117 \@ifpackageloaded{numprint}
1118   {\ifnprt@autolanguage
1119     \FBWarning{frenchb.ldf}%
1120     {Please load the "numprint" package\MessageBreak
1121      AFTER babel/frenchb; reported}%
1122   \fi
1123   }{}

```

## 2.10 Setup options: keyval stuff

All setup options are handled by command `\frenchbsetup{}` using the keyval syntax. A list of flags is defined and set to a default value which will possibly be changed ‘AtEndOfPackage’ if French is the main language. After this, `\frenchbsetup{}` eventually modifies the preset values of these flags.

Option processing can occur either in `\frenchbsetup{}`, but *only for options explicitly set* by `\frenchbsetup{}`, or ‘AtBeginDocument’; any option affecting `\extrasfrench{}` *must* be processed by `\frenchbsetup{}`: when French is the main language, `\extrasfrench{}` is executed by babel when it switches the main language and this occurs *before* reading the stuff postponed by frenchb ‘AtBeginDocument’. Reexecuting `\extrasfrench{}` is a possibility which was used up to v2.6h, it has been dropped in v3.0a because of its side-effects (f.i. `\babel@save` and `\babel@savevariable` no longer work).

`\frenchbsetup` Let’s now define this command which reads and sets the options to be processed either immediately (i.e. just after setting the key) or later (at `\begin{document}`) by `\FBprocess@options`. `\frenchbsetup{}` can only be called in the preamble.

```

1124 \newcommand*{\frenchbsetup}[1]{%
1125   \setkeys{FB}{#1}%
1126 }%
1127 \@onlypreamble\frenchbsetup

```

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

```

1128 \newif\ifFBShowOptions           \FBShowOptionsfalse
1129 \newif\ifFBStandardLayout       \FBStandardLayouttrue
1130 \newif\ifFBGlobalLayoutFrench   \FBGlobalLayoutFrenchtrue
1131 \newif\ifFBReduceListSpacing    \FBReduceListSpacingfalse

```

```

1132 \newif\ifFBListOldLayout          \FBListOldLayoutfalse
1133 \newif\ifFBCompactItemize         \FBCompactItemizefalse
1134 \newif\ifFBStandardItemizeEnv     \FBStandardItemizeEnvtrue
1135 \newif\ifFBStandardEnumerateEnv  \FBStandardEnumerateEnvtrue
1136 \newif\ifFBStandardItemLabels     \FBStandardItemLabelstrue
1137 \newif\ifFBStandardLists          \FBStandardListstrue
1138 \newif\ifFBIndentFirst            \FBIndentFirstfalse
1139 \newif\ifFBFrenchFootnotes        \FBFrenchFootnotesfalse
1140 \newif\ifFBAutoSpaceFootnotes     \FBAutoSpaceFootnotesfalse
1141 \newif\ifFBOriginalTypewriter     \FBOriginalTypewriterfalse
1142 \newif\ifFBThinColonSpace         \FBThinColonSpacefalse
1143 \newif\ifFBThinSpaceInFrenchNumbers \FBThinSpaceInFrenchNumbersfalse
1144 \newif\ifFBFrenchSuperscripts     \FBFrenchSuperscriptstrue
1145 \newif\ifFBLowercaseSuperscripts  \FBLowercaseSuperscriptstrue
1146 \newif\ifFBPartNameFull           \FBPartNameFulltrue
1147 \newif\ifFBCustomiseFigTabCaptions \FBCustomiseFigTabCaptionsfalse
1148 \newif\ifFBOldFigTabCaptions     \FBOldFigTabCaptionsfalse
1149 \newif\ifFBSmallCapsFigTabCaptions \FBSmallCapsFigTabCaptionstrue
1150 \newif\ifFBSuppressWarning        \FBSuppressWarningfalse
1151 \newif\ifFBINGuillSpace            \FBINGuillSpacefalse

```

The defaults values of these flags have been chosen 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{}`. When the beamer is loaded, lists are not customised at all to ensure compatibility.

```

1152 \edef\FB@french{\CurrentOption}
1153 \AtEndOfPackage{%
1154   \ifx\bbl@main@language\FB@french
1155     \FBGlobalLayoutFrenchtrue
1156     \@ifclassloaded{beamer}%
1157       {\PackageInfo{frenchb.ldf}{%
1158         No list customisation for the beamer class,%
1159         \MessageBreak reported}}%
1160       {\FBReduceListSpacingtrue
1161        \FBStandardItemizeEnvfalse
1162        \FBStandardEnumerateEnvfalse
1163        \FBStandardItemLabelsfalse}%
1164     \FBIndentFirsttrue
1165     \FBFrenchFootnotesttrue
1166     \FBAutoSpaceFootnotesttrue
1167     \FBCustomiseFigTabCaptionstrue
1168   \else
1169     \FBGlobalLayoutFrenchfalse
1170   \fi

```

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.

```
1171 \RequirePackage{keyval}%
1172 \define@key{FB}{ShowOptions}[true]%
1173     {\csname FBShowOptions#1\endcsname}%
1174 \define@key{FB}{StandardLayout}[true]%
1175     {\csname FBStandardLayout#1\endcsname
1176     \ifFBStandardLayout
1177         \FBReduceListSpacingfalse
1178         \FBStandardItemizeEnvtrue
1179         \FBStandardItemLabelstrue
1180         \FBStandardEnumerateEnvtrue
1181         \FBIndentFirstfalse
1182         \FBFrenchFootnotesfalse
1183         \FBAutoSpaceFootnotesfalse
1184         \FBGlobalLayoutFrenchfalse
1185     \else
1186         \FBReduceListSpacingtrue
1187         \FBStandardItemizeEnvfalse
1188         \FBStandardItemLabelsfalse
1189         \FBStandardEnumerateEnvfalse
1190         \FBIndentFirsttrue
1191         \FBFrenchFootnotesttrue
1192         \FBAutoSpaceFootnotesttrue
1193     \fi}%
1194 \define@key{FB}{GlobalLayoutFrench}[true]%
1195     {\csname FBGlobalLayoutFrench#1\endcsname
```

If this key is set to `true` when French is the main language, nothing to do: all flags keep their default value. If this key is set to `false`, nothing to do either: `\babel@save` will do the job.

```
1196         \ifFBGlobalLayoutFrench
1197             \ifx\bbbl@main@language\FB@french
1198             \else
1199                 \PackageWarning{frenchb.ldb}%
1200                 {Option 'GlobalLayoutFrench' skipped:%
1201                 \MessageBreak French is *not*
1202                 babel's last option.\MessageBreak}%
1203             \fi
1204         \fi}%
1205 \define@key{FB}{ReduceListSpacing}[true]%
1206     {\csname FBReduceListSpacing#1\endcsname}%
1207 \define@key{FB}{ListOldLayout}[true]%
1208     {\csname FBListOldLayout#1\endcsname
1209     \ifFBListOldLayout
1210         \FBStandardEnumerateEnvtrue
1211         \renewcommand*{\FrenchLabelItem}{\textendash}%
1212     \fi}%
1213 \define@key{FB}{CompactItemize}[true]%
1214     {\csname FBCompactItemize#1\endcsname
1215     \ifFBCompactItemize
1216         \FBStandardItemizeEnvfalse
```



```

1217             \FBStandardEnumerateEnvfalse
1218         \else
1219             \FBStandardItemizeEnvtrue
1220             \FBStandardEnumerateEnvtrue
1221         \fi}%
1222 \define@key{FB}{StandardItemizeEnv}[true]%
1223         {\csname FBStandardItemizeEnv#1\endcsname}%
1224 \define@key{FB}{StandardEnumerateEnv}[true]%
1225         {\csname FBStandardEnumerateEnv#1\endcsname}%
1226 \define@key{FB}{StandardItemLabels}[true]%
1227         {\csname FBStandardItemLabels#1\endcsname}%
1228 \define@key{FB}{ItemLabels}{%
1229     \renewcommand*{\FrenchLabelItem}{#1}}%
1230 \define@key{FB}{ItemLabeli}{%
1231     \renewcommand*{\Frlabelitemi}{#1}}%
1232 \define@key{FB}{ItemLabelii}{%
1233     \renewcommand*{\Frlabelitemii}{#1}}%
1234 \define@key{FB}{ItemLabeliii}{%
1235     \renewcommand*{\Frlabelitemiii}{#1}}%
1236 \define@key{FB}{ItemLabeliv}{%
1237     \renewcommand*{\Frlabelitemiv}{#1}}%
1238 \define@key{FB}{StandardLists}[true]%
1239         {\csname FBStandardLists#1\endcsname
1240         \ifFBStandardLists
1241             \FBReduceListSpacingfalse
1242             \FBCompactItemizefalse
1243             \FBStandardItemizeEnvtrue
1244             \FBStandardEnumerateEnvtrue
1245             \FBStandardItemLabelstrue
1246         \else
1247             \FBReduceListSpacingtrue
1248             \FBCompactItemizetrue
1249             \FBStandardItemizeEnvfalse
1250             \FBStandardEnumerateEnvfalse
1251             \FBStandardItemLabelsfalse
1252         \fi}%
1253 \define@key{FB}{IndentFirst}[true]%
1254         {\csname FBIndentFirst#1\endcsname}%
1255 \define@key{FB}{FrenchFootnotes}[true]%
1256         {\csname FBFrenchFootnotes#1\endcsname}%
1257 \define@key{FB}{AutoSpaceFootnotes}[true]%
1258         {\csname FBAutoSpaceFootnotes#1\endcsname}%
1259 \define@key{FB}{AutoSpacePunctuation}[true]%
1260         {\csname FBAutoSpacePunctuation#1\endcsname}%
1261 \define@key{FB}{OriginalTypewriter}[true]%
1262         {\csname FBOriginalTypewriter#1\endcsname}%
1263 \define@key{FB}{ThinColonSpace}[true]%
1264         {\csname FBThinColonSpace#1\endcsname}%
1265 \define@key{FB}{ThinSpaceInFrenchNumbers}[true]%
1266         {\csname FBThinSpaceInFrenchNumbers#1\endcsname}%
1267 \define@key{FB}{FrenchSuperscripts}[true]%

```

```

1268             {\csname FBFrenchSuperscripts#1\endcsname}
1269 \define@key{FB}{LowercaseSuperscripts}[true]%
1270             {\csname FBLowercaseSuperscripts#1\endcsname}
1271 \define@key{FB}{PartNameFull}[true]%
1272             {\csname FBPartNameFull#1\endcsname}%
1273 \define@key{FB}{CustomiseFigTabCaptions}[true]%
1274             {\csname FBCustomiseFigTabCaptions#1\endcsname}%
1275 \define@key{FB}{OldFigTabCaptions}[true]%
1276             {\csname FBOldFigTabCaptions#1\endcsname}

```

\CurrentOption no longer defined. It's value has been saved in \FB@CurOpt while reading frenchb. ldf.

```

1277         \ifFBOldFigTabCaptions
1278             \FB@addto{extras}{\babel@save\FBCaption@Separator
1279                 \def\FBCaption@Separator{\CaptionSeparator}}%
1280         \fi}%
1281 \define@key{FB}{SmallCapsFigTabCaptions}[true]%
1282     {\csname FBSmallCapsFigTabCaptions#1\endcsname
1283     \ifFBSmallCapsFigTabCaptions
1284         \let\FBfigtabshape\scshape
1285     \else
1286         \let\FBfigtabshape\relax
1287     \fi}%
1288 \define@key{FB}{SuppressWarning}[true]%
1289     {\csname FBSuppressWarning#1\endcsname
1290     \ifFBSuppressWarning
1291         \renewcommand{\FBWarning}[2]{\relax}%
1292     \fi}%

```

Here are the options controlling French guillemets spacing and the output of \frquote{ }.

```

1293 \define@key{FB}{INGuillSpace}[true]%
1294     {\csname FBINGuillSpace#1\endcsname}%
1295 \define@key{FB}{InnerGuillSingle}[true]%
1296     {\csname FBInnerGuillSingle#1\endcsname}%
1297 \define@key{FB}{EveryParGuill}{\expandafter\let\expandafter
1298     \FBEveryparguill\csname FBguill#1\endcsname}%
1299 \define@key{FB}{EveryLineGuill}{\expandafter\let\expandafter
1300     \FBEverylineguill\csname FBguill#1\endcsname
1301     \ifFB@luatex@punct
1302     \else
1303         \let\FBEverylineguill\FBguillnone
1304         \PackageWarning{frenchb. ldf}%
1305             {Option 'EveryLineGuill' skipped:%
1306             \MessageBreak this option is for
1307             LuaTeX *only*.\MessageBreak Reported}%
1308     \fi}%

```

Inputing 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. With pdfTeX (or old LuaTeX and XeTeX engines), quote characters are made active and expand to \og\ignorespaces and {\fg} respectively if the current language is

French, and to `\guillemotleft` and `\guillemotright` otherwise (think of German quotes), this is done by `\FB@@og` and `\FB@fg`; 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); the `inputenc` package has to be loaded before the `\begin{document}` with the proper coding option, so we check if `\DeclareInputText` is defined. Life is much simpler here with modern LuaTeX or XeTeX engines: we just have to activate the `\FB@addGUILspace` attribute for LuaTeX or set `\XeTeXcharclass` of quotes to the proper value for XeTeX.

```
1309 \define@key{FB}{og}{%
1310     \ifFBunicode
```

LuaTeX or XeTeX in use, first try modern LuaTeX: we just need to set LuaTeX's attribute `\FB@addGUILspace` to 1,

```
1311     \ifFB@luatex@punct
1312     \FB@addGUILspace=1 \relax
1313     \fi
```

then with XeTeX it is a bit more tricky:

```
1314     \ifFB@xetex@punct
```

`\XeTeXinterchartokenstate` is defined, we just need to set `\XeTeXcharclass` to `\FB@guilo` for the French opening quote in T1 and Unicode encoding (see subsection 2.2).

```
1315     \XeTeXcharclass"13 = \FB@guilo
1316     \XeTeXcharclass"AB = \FB@guilo
1317     \XeTeXcharclass"A0 = \FB@guilnul
1318     \XeTeXcharclass"202F = \FB@guilnul
1319     \fi
1320 \else
```

This is for conventional TeX engines:

```
1321 \newcommand*\FB@@og{%
1322     \iflanguage{french}%
1323     {\ifFBAutoSpaceGuill\FB@og\ignorespaces
1324     \else\guillemotleft
1325     \fi}%
1326     {\guillemotleft}}%
1327 \AtBeginDocument{%
1328     \ifdefined\DeclareInputText
1329     \ifdefined\uc@dclc
```

Package `inputenc` with `utf8x` encoding loaded, use `\uc@dclc`,

```
1330     \uc@dclc{171}{default}\FB@@og}%
1331     \else
```

if encoding is not `utf8x`, try `utf8`...

```
1332     \ifdefined\DeclareUnicodeCharacter
```

`utf8` loaded, use `\DeclareUnicodeCharacter`,

```
1333     \DeclareUnicodeCharacter{00AB}\FB@@og}%
1334     \else
```

if utf8 is not loaded either, we assume 8-bit character input encoding. Package MULEenc (from CJK) defines `\mule@def` to map characters to control sequences.

```

1335         \@tempcnta'#1\relax
1336         \ifdefined\mule@def
1337             \mule@def{11}{\FB@og}%
1338         \else
1339             \DeclareInputText{\the\@tempcnta}{\FB@og}%
1340         \fi
1341     \fi
1342 \fi
1343 \else

Package inputenc not loaded, no way...

1344     \PackageWarning{frenchb.ldb}%
1345         {Option 'og' requires package inputenc.\MessageBreak}%
1346     \fi
1347 }%
1348 \fi
1349 }%

```

Same code for the closing quote.

```

1350 \define@key{FB}{fg}{%
1351     \ifFBunicode
1352         \ifFB@luatex@punct
1353             \FB@addGUILSpace=1 \relax
1354         \fi
1355         \ifFB@xetex@punct
1356             \XeTeXcharclass"14 = \FB@guilf
1357             \XeTeXcharclass"BB = \FB@guilf
1358             \XeTeXcharclass"A0 = \FB@guilnul
1359             \XeTeXcharclass"202F = \FB@guilnul
1360         \fi
1361     \else
1362         \newcommand*{\FB@fg}{%
1363             \iflanguage{french}%
1364                 {\ifFBAutoSpaceGuill\FB@fg
1365                  \else\guillemotright
1366                  \fi}%
1367             {\guillemotright}}%
1368     \AtBeginDocument{%
1369         \ifdefined\DeclareInputText
1370             \ifdefined\uc@dcl
1371                 \uc@dcl{187}{default}{\FB@fg}%
1372             \else
1373                 \ifdefined\DeclareUnicodeCharacter
1374                     \DeclareUnicodeCharacter{00BB}{\FB@fg}%
1375                 \else
1376                     \@tempcnta'#1\relax
1377                     \ifdefined\mule@def
1378                         \mule@def{27}{\FB@fg}%
1379                     \else
1380                         \DeclareInputText{\the\@tempcnta}{\FB@fg}%

```

```

1381             \fi
1382         \fi
1383     \fi
1384     \else
1385         \PackageWarning{frenchb.ldf}%
1386             {Option 'fg' requires package inputenc.\MessageBreak}%
1387     \fi
1388 }%
1389 \fi
1390 }%
1391 }

```

**\FBprocess@options** \FBprocess@options will be executed at \begin{document}: it first checks about packages loaded in the preamble (possibly after babel) which customise lists: currently enumitem, paralist and enumerate; then it processes the options as set by \frenchbsetup{} or forced for compatibility with packages loaded in the preamble. When French is the main language, \extrasfrench and \captionsfrench *have already been processed* by babel at \begin{document} *before* \FBprocess@options.

```

1392 \newcommand*{\FBprocess@options}{%
    Update flags if a package customising lists has been loaded, currently: enumitem,
    paralist, enumerate.
1393 \@ifpackageloaded{enumitem}{%
1394     \ifFBStandardItemizeEnv
1395     \else
1396         \FBStandardItemizeEnvtrue
1397         \PackageInfo{frenchb.ldf}%
1398             {Setting StandardItemizeEnv=true for\MessageBreak
1399             compatibility with enumitem package,\MessageBreak}%
1400     \fi
1401     \ifFBStandardEnumerateEnv
1402     \else
1403         \FBStandardEnumerateEnvtrue
1404         \PackageInfo{frenchb.ldf}%
1405             {Setting StandardEnumerateEnv=true for\MessageBreak
1406             compatibility with enumitem package,\MessageBreak}%
1407     \fi}}%
1408 \@ifpackageloaded{paralist}{%
1409     \ifFBStandardItemizeEnv
1410     \else
1411         \FBStandardItemizeEnvtrue
1412         \PackageInfo{frenchb.ldf}%
1413             {Setting StandardItemizeEnv=true for\MessageBreak
1414             compatibility with paralist package,\MessageBreak}%
1415     \fi
1416     \ifFBStandardEnumerateEnv
1417     \else
1418         \FBStandardEnumerateEnvtrue
1419         \PackageInfo{frenchb.ldf}%
1420             {Setting StandardEnumerateEnv=true for\MessageBreak
1421             compatibility with paralist package,\MessageBreak}%

```

```

1422   \fi}{}%
1423   \@ifpackageloaded{enumerate}{%
1424     \ifFBStandardEnumerateEnv
1425     \else
1426       \FBStandardEnumerateEnvtrue
1427       \PackageInfo{frenchb.lfd}{%
1428         {Setting StandardEnumerateEnv=true for\MessageBreak
1429           compatibility with enumerate package,\MessageBreak}%
1430     \fi}{}%

```

Reset `\FB@ufl`'s normal meaning and update lists' settings in case French is the main language:

```

1431   \def\FB@ufl{\update@frenchlists}
1432   \ifx\bbbl@main@language\FB@french
1433     \update@frenchlists
1434   \fi

```

The layout of footnotes is handled at the `\begin{document}` depending on the values of flags `FrenchFootnotes` and `AutoSpaceFootnotes` (see section 2.13), 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.

```

1435   \ifBBAutoSpacePunctuation
1436     \autospace@beforeFDP
1437   \else
1438     \noautospace@beforeFDP
1439   \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.

```

1440   \ifFBOriginalTypewriter
1441   \else
1442     \let\ttfamilyORI\ttfamily
1443     \let\rmfamilyORI\rmfamily
1444     \let\sffamilyORI\sffamily
1445     \let\ttfamily\ttfamilyFB
1446     \let\rmfamily\rmfamilyFB
1447     \let\sffamily\sffamilyFB
1448   \fi

```

`ThinColonSpace` changes the normal unbreakable space typeset in French before `'` to a thin space.

```

1449   \ifFBThinColonSpace
1450     \ifFB@luatex@punct
1451       \FBcolonskip=\FBthinskip\relax
1452     \else
1453       \renewcommand*{\FBcolonspace}{\FBthinspace}%
1454     \fi
1455   \fi

```

When `true`, `INGuillSpace` resets the dimensions of skips after opening French quotes and before closing French quotes to I.N. standards.

```

1456 \ifBINGuillSpace
1457   \ifFB@luatex@punct
1458     \FBguillskip=3.33pt plus 1.665pt minus 1.11pt \relax
1459   \else
1460     \renewcommand*\FBguillspace{\space}%
1461   \fi
1462 \fi

```

When package `numprint` is loaded with option `autolanguage`, `numprint`'s command `\npstylefrench` has to be redefined differently according to the value of flag `ThinSpaceInFrenchNumbers`. As `\npstylefrench` was undefined in old versions of `numprint`, we have to provide this command.

```

1463 \@ifpackageloaded{numprint}%
1464 {\ifnprt@autolanguage
1465   \providecommand*\npstylefrench{}}%
1466   \ifBThinSpaceInFrenchNumbers
1467     \renewcommand*\npstylefrench{%
1468       \npthousandsep{\,}%
1469       \npdecimalsign{,}%
1470       \npproductsign{\cdot}%
1471       \npunitseparator{\,}%
1472       \npdegreeseperator{ }%
1473       \nppercentseparator{\nprt@unitsep}%
1474     }%
1475   \else
1476     \renewcommand*\npstylefrench{%
1477       \npthousandsep{~}%
1478       \npdecimalsign{,}%
1479       \npproductsign{\cdot}%
1480       \npunitseparator{\,}%
1481       \npdegreeseperator{ }%
1482       \nppercentseparator{\nprt@unitsep}%
1483     }%
1484   \fi
1485   \npaddtolanguage{french}{french}%
1486 \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.

```

1487 \ifBFFrenchSuperscripts
1488   \DeclareRobustCommand*\up{\@ifstar{\FB@up@fake}{\fup}}%
1489 \else
1490   \DeclareRobustCommand*\up{\@ifstar{\FB@up@fake}%
1491     {\textsuperscript}}%
1492 \fi

```

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

```

1493 \ifBLowercaseSuperscripts
1494 \else
1495 \renewcommand*\FB@lc}[1]{##1}%
1496 \fi

```

Unless `CustomiseFigTabCaptions` has been set to `false`, use `\CaptionSeparator` for koma-script, memoir and beamer classes.

```

1497 \ifBCustomiseFigTabCaptions
1498 \ifFB@koma
1499 \renewcommand*\captionformat{\CaptionSeparator}%
1500 \fi
1501 \@ifclassloaded{memoir}%
1502 {\captiondelim{\CaptionSeparator}}{}%
1503 \@ifclassloaded{beamer}%
1504 {\defbeamertemplate{caption label separator}{FBcustom}{%
1505 \CaptionSeparator}%
1506 \setbeamertemplate{caption label separator}[FBcustom]}{}%
1507 \else

```

When `CustomiseFigTabCaptions` is `false`, have the colon behave properly in French: locally force `\autospace@beforeFDP` in case of `AutoSpacePunctuation=false`.

```

1508 \ifFB@koma
1509 \renewcommand*\captionformat{{\autospace@beforeFDP : }}%
1510 \fi
1511 \@ifclassloaded{memoir}%
1512 {\captiondelim{{\autospace@beforeFDP : }}%
1513 }{}%
1514 \@ifclassloaded{beamer}%
1515 {\defbeamertemplate{caption label separator}{FBcolon}{%
1516 \autospace@beforeFDP : }}%
1517 \setbeamertemplate{caption label separator}[FBcolon]%
1518 }{}%
1519 \fi

```

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

```

1520 \ifBShowOptions
1521 \GenericWarning{* }{%
1522 * **** List of possible options for frenchb ****\MessageBreak
1523 [Default values between brackets when frenchb is loaded *LAST*]%
1524 \MessageBreak
1525 ShowOptions=true [false]\MessageBreak
1526 StandardLayout=true [false]\MessageBreak
1527 GlobalLayoutFrench=false [true]\MessageBreak
1528 StandardLists=true [false]\MessageBreak
1529 IndentFirst=false [true]\MessageBreak
1530 ReduceListSpacing=false [true]\MessageBreak
1531 ListOldLayout=true [false]\MessageBreak
1532 StandardItemizeEnv=true [false]\MessageBreak
1533 StandardEnumerateEnv=true [false]\MessageBreak
1534 StandardItemLabels=true [false]\MessageBreak
1535 ItemLabels=\textendash, \textbullet,
1536 \protect\ding{43},... [\textendash]\MessageBreak

```



```

1537 ItemLabeli=\textendash, \textbullet,
1538 \protect\ding{43},... [\textendash]\MessageBreak
1539 ItemLabelii=\textendash, \textbullet,
1540 \protect\ding{43},... [\textendash]\MessageBreak
1541 ItemLabeliii=\textendash, \textbullet,
1542 \protect\ding{43},... [\textendash]\MessageBreak
1543 ItemLabeliv=\textendash, \textbullet,
1544 \protect\ding{43},... [\textendash]\MessageBreak
1545 FrenchFootnotes=false [true]\MessageBreak
1546 AutoSpaceFootnotes=false [true]\MessageBreak
1547 AutoSpacePunctuation=false [true]\MessageBreak
1548 OriginalTypewriter=true [false]\MessageBreak
1549 ThinColonSpace=true [false]\MessageBreak
1550 ThinSpaceInFrenchNumbers=true [false]\MessageBreak
1551 FrenchSuperscripts=false [true]\MessageBreak
1552 LowercaseSuperscripts=false [true]\MessageBreak
1553 PartNameFull=false [true]\MessageBreak
1554 SuppressWarning=true [false]\MessageBreak
1555 CustomiseFigTabCaptions=false [true]\MessageBreak
1556 OldFigTabCaptions=true [false]\MessageBreak
1557 SmallCapsFigTabCaptions=false [true]\MessageBreak
1558 INGuillSpace=true [false]\MessageBreak
1559 InnerGuillSingle=true [false]\MessageBreak
1560 EveryParGuill=open, close, none [open]\MessageBreak
1561 EveryLineGuill=open, close, none
1562 [open in LuaTeX, none otherwise]\MessageBreak
1563 og= <left quote character>, fg= <right quote character>%
1564 \MessageBreak
1565 *****%
1566 \MessageBreak\protect\frenchbsetup{ShowOptions}}
1567 \fi
1568 }

```

At `\begin{document}`, we have to provide an `\xspace` command in case the `xspace` package is not loaded, do some setup for `hyperref`'s bookmarks, execute `\FBprocess@options`, switch `LuaTeX` punctuation on and issue some warnings if necessary.

```

1569 \AtBeginDocument{%
1570 \providecommand*\xspace{\relax}%

```

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

```

1571 \ifdefined\pdfstringdefDisableCommands
1572 \pdfstringdefDisableCommands{%
1573 \let\up\relax
1574 \let\fu\relax
1575 \let\degre\textdegree
1576 \let\degres\textdegree
1577 \def\ieme{e\xspace}%
1578 \def\iemes{es\xspace}%
1579 \def\ier{er\xspace}%
1580 \def\iers{ers\xspace}%

```

```

1581     \def\iere{re\space}%
1582     \def\ieres{res\space}%
1583     \def\FrenchEnumerate#1{#1\degre\space}%
1584     \def\FrenchPopularEnumerate#1{#1\degre)\space}%
1585     \def\No{N\degre\space}%
1586     \def\no{n\degre\space}%
1587     \def\Nos{N\degre\space}%
1588     \def\nos{n\degre\space}%
1589     \def\FB@og{\guillemotleft\space}%
1590     \def\FB@fg{\space\guillemotright}%
1591     \def\at{@}%
1592     \def\circonflexe{\string^}%
1593     \def\tild{\string~}%
1594     \let\bsc\textsc
1595   }%
1596 \fi

```

It is time to process the options set with `\frenchbsetup{}` or later.

```
1597 \FBprocess@options
```

With LuaTeX engines (`\FBthinskip` and `\FBcolonskip` values are set now), it is time to load file `frenchb.lua`.

```

1598 \ifFB@luatex@punct
1599   \activate@luatexpunct
1600 \fi

```

Some warnings are issued when output font encodings are not properly set. With XeLaTeX or LuaLaTeX, `fontspec.sty` and `xunicode.sty` should be loaded unless T1 encoded fonts are used through `luainputenc`, in the latter case `\FB@og` and `\FB@fg` have to be redefined; 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!

```

1601 \ifFBunicode
1602   \ifdefined\DeclareUTFcharacter
1603   \else
1604     \@ifpackageloaded{luainputenc}{}%
1605     {\PackageWarning{frenchb.ldf}%
1606       {Add \protect\usepackage{fontspec} to the\MessageBreak
1607         preamble of your document,}%
1608     }%
1609   \fi
1610 \else
1611   \begingroup \newcommand{\FBOTone}{OT1}%
1612   \ifx\encodingdefault\FBOTone
1613     \PackageWarning{frenchb.ldf}%
1614     {OT1 encoding should not be used for French.%
1615       \MessageBreak
1616       Add \protect\usepackage[T1]{fontenc} to the
1617       preamble\MessageBreak of your document,}%
1618   \fi
1619   \endgroup
1620 \fi

```

1621 }

## 2.11 French lists

`\listFB` Vertical spacing in lists should be shorter in French texts than the defaults provided by  $\LaTeX$ . Note that the easy way, just changing values of vertical spacing parameters `\listORI` 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.

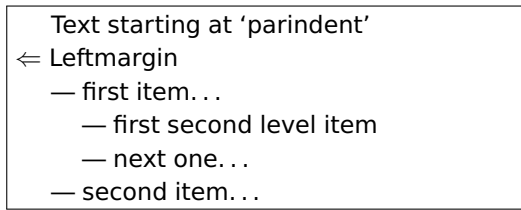
```
1622 \let\listORI\list
1623 \let\endlistORI\endlist
1624 \def\FB@listVsettings{%
1625     \setlength{\itemsep}{0.4ex plus 0.2ex minus 0.2ex}%
1626     \setlength{\parsep}{0.4ex plus 0.2ex minus 0.2ex}%
1627     \setlength{\topsep}{0.8ex plus 0.4ex minus 0.4ex}%
1628     \setlength{\partopsep}{0.4ex 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`.

```
1629     \@tempdima=\parskip
1630     \addtolength{\topsep}{-\@tempdima}%
1631     \addtolength{\partopsep}{\@tempdima}%
1632 }
1633 \def\listFB#1#2{\listORI{#1}\FB@listVsettings #2}}
1634 \let\endlistFB\endlist
```

Let's now consider French `itemize`-lists. They differ from those provided by the standard  $\LaTeX 2_{\epsilon}$  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 2.10).
- 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:



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

```

\FrenchLabelItemi1635 \newcommand*\FrenchLabelItem{\textendash}
\FrenchLabelItemii1636 \newcommand*\FrenchLabelItemii{\FrenchLabelItem}
\FrenchLabelItemiii1637 \newcommand*\FrenchLabelItemiii{\FrenchLabelItem}
\FrenchLabelItemiv1638 \newcommand*\FrenchLabelItemiv{\FrenchLabelItem}
1639 \newcommand*\FrenchLabelItemv{\FrenchLabelItem}

```

`\listindentFB` Let's define two lengths `\listindentFB` and `\labelwidthFB` to customise lists' horizontal indentations. They are given silly values here (-1pt) 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.

```

1640 \newlength\listindentFB
1641 \setlength{\listindentFB}{-1pt}
1642 \newlength\labelwidthFB
1643 \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.

```

1644 \newlength\leftmarginFB
1645 \def\FB@listHsettings{%
1646   \leftmarginFB\labelwidthFB
1647   \advance\leftmarginFB \labelsep
1648   \leftmarginii\leftmarginFB
1649   \advance\leftmarginii \listindentFB
1650   \leftmarginiii\leftmarginFB
1651   \leftmarginiiii\leftmarginFB
1652   \leftmarginiv\leftmarginFB
1653   \leftmargin\cename leftmargin\romannumeral\the\@listdepth\endcename
1654 }

```

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

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

```

1655 \def\FB@itemizesettings{%
1656   \ifFBReduceListSpacing
1657     \setlength{\itemsep}{\z@}%
1658     \setlength{\parsep}{\z@}%
1659     \setlength{\topsep}{\z@}%

```

```

1660     \setlength{\partopsep}{\z@}%
1661     \@tempdima=\parskip
1662     \addtolength{\topsep}{-\@tempdima}%
1663     \addtolength{\partopsep}{\@tempdima}%
1664 \fi
1665 \settowidth{\labelwidth}{\csname\@itemitem\endcsname}%
1666 \ifFBListOldLayout
1667     \setlength{\leftmargin}{\labelwidth}%
1668     \addtolength{\leftmargin}{\labelsep}%
1669     \addtolength{\leftmargin}{\parindent}%
1670 \else
1671     \FB@listHsettings
1672 \fi
1673 }

```

The definition of `\itemizeFB` follows the one of `\itemize` in standard  $\text{\LaTeX} 2_{\epsilon}$  classes (see `ltxlists.dtx`), spaces are customised by `\FB@itemizesettings`.

```

1674 \def\itemizeFB{%
1675     \ifnum \@itemdepth >\thr@@\@toodeep\else
1676         \advance\@itemdepth\@ne
1677         \edef\@itemitem{labelitem\romannumeral\the\@itemdepth}%
1678         \expandafter
1679         \listORI
1680         \csname\@itemitem\endcsname
1681         \FB@itemizesettings
1682     \fi
1683 }
1684 \let\enditemizeFB\endlistORI

1685 \def\labelitemsFB{%
1686     \let\labelitemi\Frlabelitemi
1687     \let\labelitemii\Frlabelitemii
1688     \let\labelitemiii\Frlabelitemiii
1689     \let\labelitemiv\Frlabelitemiv
1690     \ifdim\labelwidthFB<\z@
1691         \settowidth{\labelwidthFB}{\FrenchLabelItem}%
1692     \fi
1693     \ifdim\listindentFB<\z@
1694         \ifdim\parindent=\z@
1695             \setlength{\listindentFB}{1.5em}%
1696         \else
1697             \setlength{\listindentFB}{\parindent}%
1698         \fi
1699     \fi
1700 }

```

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

```

1701 \def\enumerateFB{%

```

```

1702 \ifnum \@enumdepth >\thr@@\@toodeep\else
1703   \advance\@enumdepth\@ne
1704   \edef\@enumctr{enum\romannumeral\the\@enumdepth}%
1705   \expandafter
1706   \list
1707     \csname label\@enumctr\endcsname
1708     {\FB@listHsettings
1709       \usecounter\@enumctr\def\makelabel##1{\hss\llap{##1}}}%
1710   \fi
1711 }
1712 \let\endenumerateFB\endlistORI

```

**\descriptionFB** Same tuning for the description environment (see the original definition in classes.dtx). Customisable `\listindentFB` added to `\itemindent` (first level only).

```

1713 \def\descriptionFB{%
1714   \list{}{\FB@listHsettings
1715     \labelwidth\z@
1716     \itemindent-\leftmargin
1717     \ifnum\@listdepth=1
1718       \advance\itemindent by \listindentFB
1719     \fi
1720     \let\makelabel\descriptionlabel}%
1721 }
1722 \let\enddescriptionFB\endlistORI

```

**\update@frenchlists** `\update@frenchlists` will set up lists according to the options of `\frenchbsetup{}`.

```

\bbbl@frenchlistlayout 1723 \def\update@frenchlists{%
\bbbl@nonfrenchlistlayout 1724 \ifFBReduceListSpacing \let\list\listFB \fi
1725 \ifFBStandardItemizeEnv
1726 \else \let\itemize\itemizeFB \fi
1727 \ifFBStandardItemLabels
1728 \else \labelitemsFB \fi
1729 \ifFBStandardEnumerateEnv
1730 \else \let\enumerate\enumerateFB \let\description\descriptionFB \fi
1731 }

```

In order to ensure compatibility with packages customising lists, the command `\update@frenchlists` should not be included in `\extrasfrench` yet, so we also define `\FB@ufl` as `\relax`, it will be redefined as `\update@frenchlists` in due time 'AtBeginDocument' by `\FBprocess@options`, see p. 54.

```

1732 \def\FB@ufl{\relax}
1733 \def\bbbl@frenchlistlayout{%
1734   \ifFBGlobalLayoutFrench
1735   \else
1736     \babel@save\list          \babel@save\itemize
1737     \babel@save\enumerate    \babel@save\description
1738     \babel@save\labelitemi   \babel@save\labelitemii
1739     \babel@save\labelitemiii \babel@save\labelitemiv
1740   \fi
1741   \FB@ufl

```

```

1742 }
1743 \def\bbl@nonfrenchlistlayout{%
1744   \ifFBGlobalLayoutFrench
1745     \update@frenchlists
1746   \fi
1747 }
1748 \FB@addto{extras}{\bbl@frenchlistlayout}
1749 \FB@addto{noextras}{\bbl@nonfrenchlistlayout}

```

## 2.12 French indentation of sections

`\bbl@frenchindent` In French the first paragraph of each section should be indented, this is another 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.

```

1750 \def\bbl@frenchindent{%
1751   \ifFBGlobalLayoutFrench\else\babel@save\@afterindentfalse\fi
1752   \ifFBIndentFirst
1753     \let\@afterindentfalse\@afterindenttrue
1754     \@afterindenttrue
1755   \fi}
1756 \def\bbl@nonfrenchindent{%
1757   \ifFBGlobalLayoutFrench
1758     \ifFBIndentFirst
1759       \@afterindenttrue
1760     \fi
1761   \fi}
1762 \FB@addto{extras}{\bbl@frenchindent}
1763 \FB@addto{noextras}{\bbl@nonfrenchindent}

```

## 2.13 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 2.10). 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` (the definition of which 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.

```

1764 \AtBeginDocument{\@ifpackageloaded{bigfoot}%
1765                   {\PackageInfo{frenchb.ldf}%
1766                    {bigfoot package in use.\MessageBreak
1767                    frenchb will NOT customise footnotes;}\MessageBreak

```

```

1768         reported}}%
1769     {\let\@footnotemarkORI\@footnotemark
1770     \def\@footnotemarkFB{\leavevmode\unskip\unkern
1771         \,\@footnotemarkORI}%
1772     \ifFBAutoSpaceFootnotes
1773     \let\@footnotemark\@footnotemarkFB
1774     \fi}%
1775 }

```

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).

```

1776 \newcommand*\dotFFN{.}
1777 \newcommand*\kernFFN{\kern .5em}
1778 \newdimen\parindentFFN
1779 \parindentFFN=10in
1780 \def\ftnISSymbol{\@fnsymbol\c@footnote}
1781 \long\def\@makefntextFB#1{\ifx\thefootnote\ftnISSymbol
1782     \@makefntextORI{#1}%
1783     \else
1784     \parindent=\parindentFFN
1785     \rule\z@\footnotesep
1786     \setbox\@tempboxa\hbox{\@thefnmark}%
1787     \ifdim\wd\@tempboxa>\z@
1788     \llap{\@thefnmark}\dotFFN\kernFFN
1789     \fi #1
1790     \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).

```

1791 \AtBeginDocument{\@ifpackageloaded{bigfoot}{}}%
1792     {\ifdim\parindentFFN<10in
1793     \else
1794     \parindentFFN=\parindent
1795     \ifdim\parindentFFN<1.5em \parindentFFN=1.5em \fi
1796     \fi
1797     \let\@makefntextORI\@makefntext
1798     \long\def\@makefntext#1{%
1799     \ifFBFrenchFootnotes
1800     \@makefntextFB{#1}%
1801     \else
1802     \@makefntextORI{#1}%

```



```

1803             \fi}%
1804         }%
1805     }

```

For compatibility reasons, we provide definitions for the commands dealing with the layout of footnotes in frenchb version 1.6. `\frenchbsetup{}` (see in section 2.10) 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 `\@makefnctext`.

```

1806 \newcommand*\AddThinSpaceBeforeFootnotes{\FBAutoSpaceFootnotestru}
1807 \newcommand*\FrenchFootnotes{\FBFrenchFootnotestru}
1808 \newcommand*\StandardFootnotes{\FBFrenchFootnotesfalse}

```

## 2.14 Clean up and exit

Final cleaning. The macro `\ldf@finish` takes care for setting the main language to be switched on at `\begin{document}` and resetting the category code of `@` to its original value. `\loadlocalcfg` is redefined locally in order not to load any `.cfg` file for French.

```

1809 \FBclean@on@exit
1810 \let\FB@llc\loadlocalcfg
1811 \let\loadlocalcfg@gobble
1812 \ldf@finish\CurrentOption
1813 \let\loadlocalcfg\FB@llc

```

### 3 Change History

v2.0	General: <code>\parindentFFN</code> not changed if already defined (required by JA for <code>cah-gut.cls</code> ). . . . . 64	v2.0g	<code>\frenchbsetup</code> : 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> . . . 46
	Added warning for OT1 encoding. . 57		
	Footnotes are now printed by default ‘à la française’ for the whole document. . . . . 63		
	New command <code>\frenchbsetup</code> added for global customisation. . 46		
	<code>\bsc</code> : <code>\hbox</code> dropped, replaced by <code>\kern0pt</code> . . . . . 37		
	<code>\captionsfrench</code> : ‘Fig.’ changed to ‘Figure’ and ‘Tab.’ to ‘Table’. . . . 40	v2.1a	General: Command <code>\fup</code> added to produce better superscripts than <code>\textsuperscript</code> . . . . . 35
	<code>\datefrench</code> : 2 ‘ <code>\relax</code> ’ added in <code>\today</code> ’s definition. . . . . 34		<code>\datefrench</code> : <code>\today</code> changed (correction in 2.0 was wrong: <code>\today</code> was printed without spaces in toc). 34
	<code>\FBtextellipsis</code> : Added special case for LY1 encoding, see bug report from Bruno Voisin (2004/05/18). 45		<code>\frenchbsetup</code> : New option: French-Superscripts to define <code>\up</code> as <code>\fup</code> or as <code>\textsuperscript</code> . . . . . 46
	<code>\nombre</code> : <code>\nombre</code> now requires <code>numprint.sty</code> . . . . . 39		New option: LowercaseSuperscripts. . . . . 46
v2.0b	General: Footnotes: Just do nothing (except warning) when the <code>bigfoot</code> package is loaded. . . . . 63	v2.1b	General: Disable some commands in bookmarks. . . . . 57
v2.0c	General: There is no need to define here <code>numprint</code> ’s command <code>\npstylefrench</code> , it will be re-defined ‘ <code>AtBeginDocument</code> ’ by <code>\FBprocess@options</code> . . . . . 40		<code>\fup</code> : Command <code>\fup</code> changed to use real superscripts from <code>fourier v. 1.6</code> . . . . . 35
	<code>\frenchbsetup</code> : Option <code>ThinSpaceInFrenchNumbers</code> added. . . . . 46	v2.1c	General: Added commands <code>\Nos</code> and <code>\nos</code> . . . . . 37
v2.0d	<code>\frenchbsetup</code> : Options <code>og</code> and <code>fg</code> changed: limit the definition to French so that quote characters can be used in German. . . . . 46		<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). . . . . 38
v2.0e	<code>\frenchbsetup</code> : New option: <code>StandardLists</code> . . . . . 46		<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). . . . . 35
v2.0f	<code>\frenchbsetup</code> : <code>StandardLayout</code> option had no effect on lists. Test moved to <code>\FBprocess@options</code> . 46	v2.1d	General: Argument of <code>\ProvidesLanguage</code> changed above from ‘ <code>french</code> ’ to ‘ <code>frenchb</code> ’ (otherwise <code>\listfiles</code> prints no date/version information). The real name of current language
	Two typos corrected in option <code>StandardLists</code> : <code>[false]</code> → <code>[true]</code> and <code>StandardLayout</code> → <code>StandardLists</code> . 46		

	(french) as to be corrected before calling <code>\LdfInit</code> . . . . .	13		
	Avoid warning “\end occurred when <code>\ifx ... incomplete</code> ” with LaTeX-2.09. . . . .	13	v2.3d	<code>\rmfamily</code> and <code>\sffamily</code> have to be robust. Bug introduced in 2.3a, pointed out by Manuel Pégourié-Gonnard. . . . .
v2.2a	<code>\frenchbsetup</code> : Default values of flags changed: default now means ‘StandardLayout’, they will be changed to ‘FrenchLayout’ AtEndOfPackage only if french is <code>\bbl@main@language</code> . . . . .	46		<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. . . . .
	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. . . . .	46	v2.3e	General: Execute <code>\AutoSpaceBeforeFDP</code> also in LaTeX to define <code>\FDP@colonspace</code> : needed for tex4ht, pointed out by MPG. . . . .
	When frenchb is babel’s last option, French becomes the document’s main language, so GlobalLayout-French applies. . . . .	46	v2.4a	General: <code>\PackageWarning</code> changed to <code>\FBWarning</code> (when bigfoot package in use). . . . .
	<code>\fup</code> : <code>\newif</code> and <code>\newdimen</code> moved before <code>\ifLaTeXe</code> to avoid an error with plainTeX. . . . .	35		<code>\CaptionSeparator</code> : <code>\PackageWarning</code> changed to <code>\FBWarning</code> (in case <code>\@makecaption</code> has been customised). <code>\FBWarning</code> is defined as <code>\PackageWarning</code> by default but can be made silent using <code>\frenchbsetup</code> , (suggested by MPG). . . . .
v2.3a	General: <code>\NoAutoSpaceBeforeFDP</code> and <code>\AutoSpaceBeforeFDP</code> now set the flag <code>\ifFBAutoSpacePunctuation</code> accordingly (LaTeX only). . . . .	28		<code>\frenchbsetup</code> : New option SuppressWarning. . . . .
	In LaTeX, frenchb no longer adds spaces before ‘high punctuation’ characters in computer code. Suggested by Yannis Haralambous. . . . .	29		<code>\ifFBXeTeX</code> : Added a new ‘if’ <code>\FBunicode</code> and some <code>\lccode</code> definitions to <code>\extrasfrench</code> and <code>\noextrasfrench</code> . . . . .
	<code>\frenchbsetup</code> : New option: OriginalTypewriter. Now frenchb switches to <code>\noautospace@beforeFDP</code> when a tt-font is in use. When OriginalTypewriter is set to true, frenchb behaves as in pre-2.3 versions. . . . .	46	v2.4c	General: In <code>\ttfamilyFB</code> , also cancel automatic spaces inside French guillemets coded as characters (see <code>\frenchbsetup</code> ). . . . .
	<code>\fup</code> : <code>\lowercase</code> changed to <code>\MakeLowercase</code> as the former doesn’t work for non ASCII characters in encodings like applemac, utf-8, . . . . .	35		In <code>\ttfamilyFB</code> , also cancel automatic spaces inside French guillemets entered as characters (see <code>\frenchbsetup</code> ). . . . .
v2.3b	General: New commands <code>\dotFFN</code> and <code>\kernFFN</code> for more flexibility (suggested by JA). . . . .	64	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 (catalan.lfd). Bug pointed out by Felip Manyé i Ballester. . . . .
v2.3c	General: Commands <code>\ttfamily</code> ,			

v2.5a	General: <code>\og</code> and <code>\fg</code> do not print correctly in English when using XeTeX or LuaTeX, fixed by using <code>\textquotedblleft</code> and <code>\textquotedblright</code> defined above. ....	32	the status of the French “apostrophe”. ....	15
	New command <code>\NoAutoSpacing</code> , suggested by MPG. ....	30		
	Punctuation is no longer made active with XeTeX-based engines. ....	16		
	<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. ....	40		
	<code>\FBthinspace</code> : Define <code>\FBthinspace</code> for those who want to customise the width of the space before ; and co. ....	17		
	<code>\textquoteddblright</code> : Change <code>\guillemotleft</code> and <code>\guillemotright</code> definitions for Unicode anf provide definitions for <code>\textquotedblleft</code> and <code>\textquotedbright</code> . Insures correct printing of quotes by <code>\og</code> and <code>\fg</code> in French and outside. ....	30		
v2.5b	General: 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. ....	47		
v2.5c	General: The code meant for XeTeX also works for LuaTeX, we just need to change the test. ....	51		
v2.5d	General: Moved the <code>\newcount</code> command outside <code>\ifFB@xetex@punct ... \fi</code> (it broke Plain formats). ..	24		
	<code>\ifFBXeTeX</code> : Added two new ‘if’ <code>\FBXeTeX</code> and <code>\FBLuaTeX</code> as XeTeX and behave differently regarding			
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. ....	57		
v2.5f	General: Changed definitions of <code>\at</code> , <code>\circflexe</code> , <code>\tild</code> , <code>\boi</code> and <code>\degre</code> for Unicode based engines. ....	37		
	<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). ....	45		
v2.5g	General: Redefine <code>\degre</code> , <code>\degres</code> <code>\at</code> <code>\circflexe</code> and <code>\tild</code> for bookmarks. Add <code>\fup</code> also. ....	57		
	When <code>\ifFB@xetex@punct</code> is true, ‘og’ and ‘fg’ options now set XeTeX-charclasses of these characters to <code>\FB@guilo</code> and <code>\FB@guilf</code> . Otherwise French quotes behave as normal characters (their XeTeXcharclass is 0). ....	51		
	<code>\FB@xetex@punct@french</code> : XeTeX-charclass(es) for French quotes will be set to <code>\FB@guilo</code> and <code>\FB@guilf</code> by options ‘og’ and ‘fg’ in <code>\frenchbsetup</code> . French quotes should behave as normal characters by default in XeLaTeX as in LaTeX. ....	25		
v2.5h	<code>\degres</code> : <code>textcomp.sty</code> has changed. The test about <code>\M@TS1</code> is no longer relevant, let’s change it. ....	38		
v2.5i	General: Temporary fix: as long as <code>xeCJK.sty</code> will not use <code>\newXeTeXintercharclass</code> to allocate its classes, we will have to define 3 fake classes. ....	24		
	<code>\FB@xetex@punct@french</code> : <code>xeCJK.sty</code> changes the <code>\XeTeXcharclass</code> of ASCII chars ‘ ’ ’ ’ ’ ’ ’ ’ ’ ’			

'{' '%' opening and closing single and double quotes. We set their class to 0 in French and reset their class to their original value when leaving French. See <code>\FB@xetex@punct@nonfrench</code> below. ....	25	<code>\FBthinspace</code> and <code>\Fcolonspace</code> to <code>\FBcolonspace</code> to avoid a conflict with <code>fournier.sty</code> . ....	17
v2.5j	General: Previous fix removed: bug fixed in <code>xeCJK.sty</code> version 3.0.4 (06-May-2012). ....	v2.6e	<code>\degres</code> : Refrain from redefining <code>\textdegree</code> from <code>latin1.def</code> , <code>applemac.def</code> , etc. as <code>\degres</code> because it loops in <code>hyperref</code> 's bookmarks. Pointed out by Eddy Flas on <code>fctt</code> . ....
v2.6a	General: Bug correction: changing <code>\leftmargin</code> cannot be done only for <code>itemize</code> -lists: it messes up embedded <code>enumerate</code> lists. Pointed out by Denis Bitouzé. Lists have been completely redesigned in <code>frenchb v. 2.6a</code> . An option for backward compatibility is provided. .	v2.6f	<code>\FB@itemizesettings</code> : <code>\labelwidth</code> must be reset, f.i. when an <code>itemize</code> list occurs inside environments based on <code>trivlist</code> which set <code>\labelwidth</code> to 0 (see proof environment in <code>amsthm.sty</code> ). Bug pointed out by Julien Hauseux. . .
	<code>\frenchbsetup</code> : New options <code>ListOldLayout</code> , <code>StandardItemizeEnv</code> and <code>StandardEnumerateEnv</code> ( <code>CompactItemize</code> is deprecated). ....	v2.6g	General: <code>U+00A0</code> (Unicode nobreakspace) and <code>U+202F</code> (Unicode nobreakthinspace) added to class <code>\FB@punctnul</code> to prevent <code>frenchb</code> from adding it's own space before 'high punctuation' characters. . .
	<code>\FrenchLabelItem</code> : default changed from <code>\textendash</code> to <code>\textemdash</code> . ....		<code>\FB@itemizesettings</code> : Suppress all vertical spaces only if <code>ReduceListSpacing</code> is true. Pointed out by Pierre Willaime. ....
v2.6b	<code>\descriptionFB</code> : Settings of <code>\FB@listHsettings</code> should apply to <code>description</code> lists too. ....		<code>\ifFBXeTeX</code> : <code>lccode</code> values for the French "apostrophe" are now the same for <code>XeTeX</code> and <code>LuaTeX</code> . ....
v2.6c	General: Dummy file <code>frenchb.cfg</code> is no longer generated from <code>frenchb.dtx</code> . ....	v2.6h	General: <code>\FG@og</code> and <code>\FG@fg</code> changed: former clumsy code removed. . .
	No warning about <code>\makecaption</code> for AMS classes. ....		If <code>\makecaption</code> is undefined, no warning. ....
	No warning about <code>\makecaption</code> for <code>koma-script</code> classes. <code>\captionformat</code> customised in French. ....		New class <code>\FB@guilnul</code> for characters <code>U+00A0</code> (Unicode nobreakspace) and <code>U+202F</code> (Unicode nobreakthinspace), to prevent <code>frenchb</code> from adding spurious spaces inside quotes. ....
	Warning added when the <code>caption</code> or <code>floatrow</code> package is loaded before <code>babel/frenchb</code> . ....		<code>\CaptionSeparator</code> : No active catcodes in <code>\STD@makecaption</code> 's definition. ....
	<code>\CaptionSeparator</code> : Former <code>\CaptionSeparator</code> has been renamed as <code>\FBCaption@Separator</code> ; Newif <code>\if@FBwarning@capsep</code> added. ....	v3.0a	General: <code>\LdfInit</code> checks <code>\datefrench</code> instead of <code>\captionfrench</code> to avoid a conflict with <code>paper.tex.cls</code> which loads
v2.6d	<code>\FBthinspace</code> : Rename <code>\Fthinspace</code>		

datetime.sty. ....	13	\extrasfrench: Take advantage of babel's \babel@savevariable to handle apostrophe's \lccode. . .	15
\bbl@nonfrenchguillemets deleted, use \babel@save instead. ....	32	\FBprocess@options: Changed option ThinColonSpace to make it work also with LuaTeX. ....	54
Added explicit \FBguillskip for LuaTeX. ....	31	With koma-script and memoir class, customise \captionformat and \captiondelim. ....	56
Definitions of \FB@og and \FB@fg now depend on punctuation handling (LuaTeX / XeTeX / active). .	31	\FBthinspace: LuaTeX requires dimensions: two new skips \FBcolonskip and \FBthinspace. .	17
french.cfg will be loaded (if found) instead of frenchb.cfg. NO NEED for .cfg files in French anyway. . .	65	\frenchbsetup: New options OldFigTabCaptions and CustomiseFigTabCaptions. ....	46
In Plain, provide a substitute for \PackageWarning and \PackageInfo. ....	14	v3.0b	
Merging of \captionsfrenchb, \captionsfrançais with \captionsfrench deleted in favor of new babel 3.9 syntax. ....	42	General: frenchb.lua was not found by Lua function dofile (not kpathsea aware). Call function kpse.find_file first, as suggested by Paul Gaborit. ....	23
More informative, less TeXnical warning about \@makecaption. .	44	Require luatexbase with LaTeX in case fontspec has not been loaded before babel. ....	17
New flag \ifFB@luatex@punct for 'high punctuation' management with LuaTeX engines. ....	16	v3.0c	
New handling of 'high punctuation' through callbacks with LuaTeX engines. ....	17	General: Activate option StandardLists when beamer class is loaded. . .	47
No warning about \@makecaption for SMF classes. No warning either with LuaTeX or XeTeX engines. . .	43	Changed \FBguill@spacing (internal) to \FBguillspace (public). .	31
Options processing completely reorganised. ....	46	frenchb requires babel-3.9i. ....	14
Support for options frenchb, français, canadien, acadian changed. .	13	frenchb.lua: null glues should not trigger space insertion before high punctuation. Bug pointed out by Benoit Rivet for the 'lstlisting' environment of the listings package. .	20
Test \ifXeTeX changed to \ifFBunicode and 'xltextra' changed to 'fontspec'. ....	58	Just load luatexbase.sty instead of luaotfloat.sty with plain formats. .	17
\CaptionSeparator: Remove \CaptionSeparatorORI, use \babel@save instead. ....	42	No need to define \@french as \lang@french, babel.def (3.9j) takes care for this. ....	13
\captionsfrench: Take advantage of babel's \SetString commands for captionnames. ....	40	\datefrench: \SetString still does not work for Plain with babel 3.9k. Need to define \datefrench. . .	34
\datefrench: Take advantage of babel's \SetString commands for \datefrench. Doesn't work with Plain (yet?). ....	34	\frenchbsetup: New option INGuillSpace. ....	46
\descriptionFB: Add \listindentFB to \itemindent. Suggested by Denis Bitouzé. ....	62	v3.1a	
		General: Codes "13 and "14 added for French quotes in T1-encoding. Support for older versions of LuaTeX and XeTeX dropped. ....	51

fontspec is not required for T1 fonts used with the luainputenc.sty package. ....	58	v3.1f	General: \FBCaption@Separator changed when option CustomiseFigTabCaptions is set to false. . .	44	
frenchb.lua: added flag addgl which must also be true when prev or next is not a char (i.e. kern0 in «\texttt{a}»).	21	\FBprocess@options: Bug fix for the beamer class: figure and table captions are now consistent with frenchb's documentation. Pointed out by Denis Bitouzé. ....	56	Definition of \captionformat and \captiondelim changed when option CustomiseFigTabCaptions is set to false. ....	56
frenchb.lua: codes 0x13 and 0x14 added for French quotes in T1-encoding. ....	18	\FBthinspace: \FBthinspace is no longer a kern but a skip (frenchb adds a nobreak penalty before it).	17		
frenchb.lua: look ahead when next is a kern (i.e. in « \texttt{a} »).	22				
Misplaced \fi for plain formats. . .	17	v3.1g	General: frenchb.lua: flag addgl set to false for '«' at the end of an \hbox or a paragraph or when followed by a null glue (i.e. springs). . . .	22	
New command \frquote for imbedded or long French quotations. . .	32	\frenchb.lua: flag addgl set to false for '»' at the beginning of an \hbox or a paragraph or a tabular 'l' and 'c' columns. ....	21	frenchb.lua: node HLIST added; node TEMP added for the first node of \hboxes. ....	19
\frenchbsetup: New options InnerGuillSingle, EveryParGuill and EveryLineGuill to control \frquote. . .	46	Lua function french_punctuation is now inserted at the end of the "kerning" callback (no priority) instead of "hpack_filter" and "pre_linebreak_filter". ....	23	Use Babel defined loops \bb@for instead of \@for borrowed from file ltcntrl.dtx (\@for is undefined in Plain). ....	24
v3.1b		\captionsfrench: \partname's definition depends now on flag PartNameFull. No need to redefine it in \frenchbsetup. ....	40		
General: frenchb.lua: add a check for null fid in french_punctuation (Tikz \nullfont). Bug pointed out by Paul Gaborit. ....	20	Bug fix for koma-scripts classes: a spurious dot was added by the \partformat command. ....	41	\frenchbsetup: PartNameFull now just sets the flag, nothing to add to \captionsfrench when false. . .	46
\captionsfrench: Change \scshape to customisable \FBfigtabshape for \figurename and \tablename. ....	40				
\fprimo): Removed \lowercase from definitions of \FrenchEnumerate, ... \No and co: \up already does the conversion. ....	37	v3.1h	General: french.cfg from e-french conflicts with frenchb. Do NOT load it (no need for .cfg files with frenchb anyway). ....	65	
\frenchbsetup: New option SmallCapsFigTabCaptions. ....	46				
\ieres: Removed \lowercase from definitions of \ieme and co: \up already does the conversion. . . .	36				
v3.1c					
General: frenchb.lua: Previous bug fix for null glues (v3.0c) did not work properly. Fixed now (I hope). Pointed out by Jacques André. . .	20				
v3.1d					
General: New section: issue warnings if packages listings, numprint and natbib are loaded too early or too late vs babel. ....	45				
v3.1e					
\frenchbsetup: Corrected typo: SmallCapsFigTabcaptions instead of SmallCapsFigTabCaptions. Pointed out by Céline Chevalier. . .	46				