

# A Babel language definition file for French

**frenchb.dtx v3.1a, 2014/06/21**

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>	5
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	23
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 Setup options: keyval stuff	45
2.10 French lists	57
2.11 French indentation of sections	61
2.12 Formatting footnotes	62
2.13 Clean up and exit	63
<b>3 Change History</b>	<b>64</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 v3.1a 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.

---

<sup>1</sup>The file described in this section has version number v3.1a and was last revised on 2014/06/21.

<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).

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

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 ‘XeTeXinter-char’ 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 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`.

---

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

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<sup>o</sup> N<sup>os</sup> n<sup>o</sup> and n<sup>os</sup> ) are obtained via the commands `\No`, `\Nos`, `\no`, `\nos`.
6. Two commands are provided to typeset the symbol for “degré”: `\degre` prints the raw character and `\degres` should be used to typeset temperatures (e.g., “20~\degres C” with an unbreakable space), or for alcohols’ strengths (e.g., “45\degres” with *no* space in French).
7. In math mode the comma has to be surrounded with braces to avoid a spurious space being inserted after it, in decimal numbers for instance (see the T<sub>E</sub>Xbook p. 134). The command `\DecimalMathComma` makes the comma be an ordinary character *in French only* (no space added); as a counterpart, if `\DecimalMathComma` is active, an explicit space has to be added in lists and intervals: `$(0,\ 1)$`, `$(x,\ y)$`. `\StandardMathComma` switches back to the standard behaviour of the comma.
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 leftmargins' 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 options applies to all languages.

`FrenchFootnotes=false (true*)` reverts to the standard layout of footnotes.

By default frenchb typesets leading numbers as ‘1. ’ instead of ‘<sup>1</sup>’, 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 older documents. By default `\up` now relies on `\fup` designed to produce better looking superscripts.

`AutoSpacePunctuation=false (true)` ; in French, the user *should* input a space before the four characters ‘:;!?’ but as many people forget about it (even among native French writers!), the default behaviour of frenchb is to automatically typeset nobreakspaces the width of which is either `\FBthinspace` (defaults to `\thinspace`) 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 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 preceeding 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 L<sup>A</sup>T<sub>E</sub>X classes only.

`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.

For *all* languages, when French is the main language, frenchb changes the separator (colon) used in figures’ and tables’ captions 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.

Two 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.



### 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 (desactivated by default) to suppress warnings if `\@makecaption` has been redefined or if the `bigfoot` package is in use.

French hyphenation patterns are now coded in Unicode, see file `hyph-fr.tex`. Extra code has been added to deal with hyphenation of the French “apostrophe” with 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  $\text{\LaTeX-2.09}$  and for  $\text{\LaTeX 2}_{\epsilon}$  in compatibility mode has been dropped. This version is meant for  $\text{\LaTeX 2}_{\epsilon}$  and Plain based formats (like `bplain`).  $\text{\LaTeX 2}_{\epsilon}$  formats based on `\text{\mlTeX}` 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\bbl@tempa{francais}
4 \ifx\CurrentOption\bbl@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\bbl@tempa{frenchb}
13 \ifx\CurrentOption\bbl@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\bbl@nulllanguage{\string\language=0}%
26     \adddialect\l@french0
27   \else
28     \adddialect\l@french\l@nohyphenation
29     \edef\bbl@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\bbl@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\bbl@tempa\endinput
44 \else
45   \LaTeXettrue
46 \fi
47 \fi
48 \bbl@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\bbl@tempa\relax
68 \ifx\babeltags\@undefined
69   \let\bbl@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`.

**\ifBLaTeX** XeTeX and LuaTeX engines require some extra code to deal with the French “apostrophe”. Let’s define three new ‘if’: `\ifBLaTeX`, `\ifBTeX` and `\ifBUnicode`

which will be true for XeTeX and LuaTeX engines and false for 8-bits engines.

We cannot rely on  $\varepsilon$ -TeX’s `\ifdefined` at this stage, as it is not defined in Plain TeX format.

```

101 \newif\ifFBUnicode
102 \newif\ifBLaTeX
103 \newif\ifBTeX
104 \begingroup\expandafter\expandafter\expandafter\endgroup
105 \expandafter\ifx\csname luatexversion\endcsname\relax
106 \else
107   \FBUnicodetrue \BLaTeXtrue
108 \fi
109 \begingroup\expandafter\expandafter\expandafter\endgroup
110 \expandafter\ifx\csname XeTeXrevision\endcsname\relax
111 \else
112   \FBUnicodetrue \BTeXtrue
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 LuaTeX, 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, **\FBthinspace** the other three require just a `\thinspace`. We define **\FBcolonspace** as `\space` (inter-word space) and **\FBthinspace** as `\thinspace` (both are user customisable). LuaTeX requires skips instead of commands, so we define **\FBcolonskip** and **\FBthinskip** to hold the specifications (width/stretch/shrink) of `\space` and `\thinspace` for the `lmr10` 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}{\thinspace}
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 KERN             = node_id("kern")
200 local GLUE             = node_id("glue")
201 local GSPEC            = node_id("glue_spec")
202 local GLYPH            = node_id("glyph")
203 local PENALTY          = node_id("penalty")
204 local nobreak          = new_node(PENALTY)
205 nobreak.penalty        = 10000
206 local insert_node_before = node.insert_before
207 local insert_node_after  = node.insert_after
208 local remove_node        = node.remove

```

Some variables to store \FBthinskip, \FBcolonskip and \FBguillskip (given for lmr10); width/stretch/shrink are stored as fractions of \fontdimen2, \fontdimen3 and \fontdimen4 of lmr10 font respectively...

```

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

```

and a function to scale them for the current font:

```

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

```

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

```

236 local addDPspace      = luatexbase.attributes['FB@addDPspace']
237 local addGUILspace    = luatexbase.attributes['FB@addGUILspace']
238 local has_attribute    = node.has_attribute

```

The following function will be added to `pre_linebreak_filter` and `hpack_filter` callbacks. 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`).

```

239 local function french_punctuation (head)
240   for item in node.traverse_id(GLYPH, head) do
241     local lang = item.lang
242     local char = item.char
243     local SIG = has_attribute(item, addGUILspace)
244     if lang == FR and FB_punct_left[char] then
245       local fid = item.font
246       local prev = item.prev
247       local prev_id, prev_subtype, prev_char
248       if prev then
249         prev_id = prev.id
250         prev_subtype = prev.subtype
251         if prev_id == GLYPH then
252           prev_char = prev.char
253         end
254       end

```

If the previous item is a glue, check its natural width, only positive glues are to be replaced by a `nobreakspace`.

```

255       local glue = prev_id == GLUE and prev_subtype == 0
256       local glue_wd
257       if glue then
258         glue_spec = prev.spec
259         glue_wd = glue_spec.width
260       end
261       glue = glue and glue_wd > 0

```

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`, `thinst`, `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 *only if* all three conditions are satisfied: a) attribute `\FB@addDPspace` is set, b) the previous character is not part of type `FB_punct_null` (this avoids spurious spaces in strings like `( ! )` or `??`), c) the punctuation mark does not start a paragraph.

```

262       if FB_punct_thin[char] or FB_punct_thick[char] then
263         local SBDP = has_attribute(item, addDPspace)
264         local fbglue
265         if FB_punct_thick[char] then
266           fbglue = new_glue_scaled(fid, colnwd, colnst, colnsh)
267         else
268           fbglue = new_glue_scaled(fid, thinwd, thinst, thinsh)

```

```

269         end
270         local auto =
271             SBDP and SBDP > 0 and
272             ((prev_char and not FB_punct_null[prev_char]) or
273             (not prev_char and (prev_id ~= 0 or prev_subtype ~= 3))
274         )
275         if glue or auto then
276             if glue then
277                 head = remove_node(head,prev,true)
278             end
279             insert_node_before(head, item, copy_node(nobreak))
280             insert_node_before(head, item, copy_node(fbglue))
281         end

```

Let's consider '»' now (the only remaining glyph of FB\_punct\_left class): we just have to remove any *glue* possibly preceeding '»', then 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).

```

282         elseif SIG and SIG > 0 then
283             local addgl = (prev_char and not FB_guil_null[prev_char]) or
284                         (not prev_char)
285             if glue or addgl then
286                 if glue then
287                     head = remove_node(head,prev,true)
288                 end
289                 local fbglue = new_glue_scaled(fid,guilwd,guilst,guilsh)
290                 insert_node_before(head, item, copy_node(nobreak))
291                 insert_node_before(head, item, copy_node(fbglue))
292             end
293         end
294     end

```

Similarly, for '«' (unique member of the FB\_punct\_right class), we check the following node looking for *glue* or FB\_guil\_null character, in order to remove any *glue* possibly following it and to insert the proper *glue* and nobreak penalty in this order if necessary.

```

295     if lang == FR and FB_punct_right[char] and SIG and SIG > 0 then
296         local next = item.next
297         local next_id, next_subtype, next_char, kern_wd, nextnext
298         if next then
299             next_id = next.id
300             next_subtype = next_subtype
301             if next_id == GLYPH then
302                 next_char = next.char

```

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

```

303         elseif next_id == KERN then
304             kern_wd = next.kern
305             if kern_wd == 0 then

```

```

306         nextnext = next.next
307         if nextnext then
308             next = nextnext
309             next_id = nextnext.id
310             next_subtype = nextnext.subtype
311             next_char = nextnext.char
312         end
313     end
314 end
315 end
316 local glue = next_id == GLUE and next_subtype == 0
317 if glue then
318     glue_spec = next.spec
319     glue_wd = glue_spec.width
320 end
321 glue = glue and glue_wd > 0
322 local addgl = (next_char and not FB_guil_null[next_char]) or
323             (not next_char)
324 if glue or addgl then
325     if glue then
326         head = remove_node(head, next, true)
327     end
328     local fid = item.font
329     local fbglue = new_glue_scaled(fid, guilwd, guilst, guilsh)
330     insert_node_after(head, item, copy_node(fbglue))
331     insert_node_after(head, item, copy_node(nobreak))
332 end
333 end
334 end
335 return head
336 end
337 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.

```

338 \ifFB@luatex@punct
339   \newcommand*{\FB@luatex@punct@french}{%
340     \ifx\shorthandoffORI\undefined
341       \let\shorthandonORI\shorthandon
342       \let\shorthandoffORI\shorthandoff
343     \fi
344     \def\shorthandoff##1{%
345       \ifx\PackageWarning\undefined
346         \fb@warning{\noexpand\shorthandoff{;:!?} is helpless with
347           LuaTeX,\, use \noexpand\NoAutoSpacing
348           *inside a group* instead.}%
349       \else
350         \PackageWarning{frenchb.ldf}{\protect\shorthandoff{;:!?} is
351           helpless with LuaTeX,\MessageBreak use \protect\NoAutoSpacing

```

```

352         \space *inside a group* instead;\MessageBreak reported}%
353     \fi}%
354     \def\shorthandon##1{%
355 }
356 \newcommand*{\FB@luatex@punct@nonfrench}{%
357     \ifx\shorthandoffORI\undefined
358     \else
359         \let\shorthandon\shorthandonORI
360         \let\shorthandoff\shorthandoffORI
361     \fi
362 }
363 \FB@addto{extras}{\FB@luatex@punct@french}
364 \FB@addto{noextras}{\FB@luatex@punct@nonfrench}

```

In  $\text{\LaTeX}2_{\epsilon}$ , 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 to both callbacks pre\_linebreak\_filter (paragraph building) and hpack\_filter (\hbox building).

```

365 \def\activate@luatexpunct{%
366     \directlua{%
367         FR = \the\l@french
368         local path = kpse.find_file("frenchb.lua", "lua")
369         if path then
370             local f = dofile(path)
371             luatexbase.add_to_callback("pre_linebreak_filter",
372                                     f, "frenchb.french_punctuation",1)
373             luatexbase.add_to_callback("hpack_filter",
374                                     f, "frenchb.french_punctuation",1)
375         else
376             texio.write_nl('')
377             texio.write_nl('*****')
378             texio.write_nl('Error: frenchb.lua not found.')
379             texio.write_nl('*****')
380             texio.write_nl('')
381         end
382     }%
383 }
384 \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 2.9).

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  `; ! ? : ( ] « »` 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.

```

385 \newcount\FB@interchartokenstateORI
386 \ifFB@xetex@punct
387   \ifLaTeXe
388     \PackageInfo{frenchb.ldf}{No need for active punctuation characters%
389       \MessageBreak with this version of XeTeX!%
390       \MessageBreak reported}
391   \else
392     \fb@info{No need for active punctuation characters\\
393       with this version of XeTeX!}
394   \fi

```

The following code is borrowed from `ltnctrl.dtx` (LaTeX base) for loops (`\@for` undefined in Plain):

```

395 \ifx\@for\undefined
396   \def\@nnil{\@nil}%
397   \def\@empty{}%
398   \def\@fornoop#1\@#2#3{%
399     \long\def\@for#1:=#2\do#3{%
400       \expandafter\def\expandafter\@fortmp\expandafter{#2}%
401       \ifx\@fortmp\@empty \else
402         \expandafter\@forloop#2,\@nil,\@nil\@#1{#3}\fi}%
403     \long\def\@forloop#1,#2,#3\@#4#5{\def#4{#1}\ifx #4\@nnil \else
404       #5\def#4{#2}\ifx #4\@nnil \else#5\@iforloop #3\@#4{#5}\fi\fi}%
405     \long\def\@iforloop#1,#2\@#3#4{\def#3{#1}\ifx #3\@nnil
406       \expandafter\@fornoop \else
407       #4\relax\expandafter\@iforloop\fi#2\@#3{#4}}%
408     \def\@tfor#1:={\@tfor#1 }%
409     \long\def\@tfor#1#2\do#3{\def\@fortmp{#2}\ifx\@fortmp\space\else
410       \@tforloop#2\@nil\@nil\@#1{#3}\fi}%
411     \long\def\@tforloop#1#2\@#3#4{\def#3{#1}\ifx #3\@nnil
412       \expandafter\@fornoop \else
413       #4\relax\expandafter\@tforloop\fi#2\@#3{#4}}%
414   \fi

```

Six new character classes are defined for `frenchb`.

```

415 \newXeTeXintercharclass\FB@punctthick
416 \newXeTeXintercharclass\FB@punctthin
417 \newXeTeXintercharclass\FB@punctnul
418 \newXeTeXintercharclass\FB@guilo
419 \newXeTeXintercharclass\FB@guilf
420 \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.

```

421 \def\FB@charclassesORI{}
422 \def\empty{}
423 \def\FB@parse#1,#2\endparse{\def\FB@class{#1}%
424 \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.

```

425 \newcommand*{\FB@xetex@punct@french}{%

```

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

```

426 \ifx\FB@charclassesORI\empty
427 \FB@interchartokenstateORI=\XeTeXinterchartokenstate
428 \@for\FB@char:={'\:,\';,\!','\?', "AB,"BB,%
429 '\(,\'[,\'[{\,\',\.\,\-,\'),'\],'\},%
430 '\%, "22,"27,"60,"2019,"A0,"202F}\do
431 {\edef\FB@charclassesORI{\FB@charclassesORI%
432 \the\XeTeXcharclass\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 \@for\FB@char:={'\:,\';,\!','\?}\do
445 {\XeTeXcharclass\FB@char=\FB@punctthin}%
446 \XeTeXinterchartoks \z@ \FB@punctthin = {%
447 \ifhmode\FDP@thinspace\fi}%
448 \XeTeXinterchartoks \FB@guilf \FB@punctthin = {%
449 \FDP@thinspace}%
450 \XeTeXinterchartoks 255 \FB@punctthin = {%
451 \ifhmode\unskip\penalty\@M\FBthinspace\fi}%
452 \XeTeXinterchartoks \FB@guilo \z@ = {%
453 \ifFBAutoSpaceGuill\FBguillspace\fi}%
454 \XeTeXinterchartoks \FB@guilo 255 = {%
455 \ifFBAutoSpaceGuill\FBguillspace\ignorespaces\fi}%
456 \XeTeXinterchartoks \z@ \FB@guilf = {%
457 \ifFBAutoSpaceGuill\FBguillspace\fi}%
458 \XeTeXinterchartoks \FB@punctthin \FB@guilf = {%
459 \ifFBAutoSpaceGuill\FBguillspace\fi}%
460 \XeTeXinterchartoks 255 \FB@guilf = {%
461 \ifFBAutoSpaceGuill\unskip\FBguillspace\fi}%

```

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

```
462 \@for\FB@char:={'\[, '\[, "A0,"202F}\do
463 {\XeTeXcharclass\FB@char=\FB@punctnul}%
```

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

```
464 \@for\FB@char:={'\{, '\{, '\., '\., '\), '\), '\], '\], '\%, %
465 "22,"27,"60,"2019}\do
466 {\XeTeXcharclass\FB@char=\z@}%
```

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

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

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

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

End of specific code for punctuation with modern XeTeX engines.

```
495 \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.

```
496 \ifFB@active@punct
497   \initiate@active@char{:}%
498   \initiate@active@char{;}%
499   \initiate@active@char{!}%
500   \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.

```
501   \declare@shorthand{french}{;}{;%
502     \ifhmode
503       \ifdim\lastskip>\z@
504         \unskip\penalty\M\FBthinspace
505       \else
506         \FDP@thinspace
507       \fi
508     \fi
```

Now we can insert a ; character.

```
509     \string;}
```

The next three definitions are very similar.

```
510   \declare@shorthand{french}{!}{;%
511     \ifhmode
512       \ifdim\lastskip>\z@
513         \unskip\penalty\M\FBthinspace
514       \else
515         \FDP@thinspace
516       \fi
517     \fi
518     \string!}
519   \declare@shorthand{french}{?}{;%
520     \ifhmode
521       \ifdim\lastskip>\z@
522         \unskip\penalty\M\FBthinspace
523       \else
524         \FDP@thinspace
525       \fi
526     \fi
527     \string?}
528   \declare@shorthand{french}{:}{;%
529     \ifhmode
530       \ifdim\lastskip>\z@
531         \unskip\penalty\M\FBcolonspace
532       \else
```

```

533         \FDP@colonspace
534         \fi
535     \fi
536     \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.

```

537 \declare@shorthand{system}{:}{\string;}
538 \declare@shorthand{system}{!}{\string!}
539 \declare@shorthand{system}{?}{\string?}
540 \declare@shorthand{system}{;}{\string;}
541 %}

```

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

```

542 \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.

```

543     \bbl@activate{:}\bbl@activate{;}%
544     \bbl@activate{!}\bbl@activate{?}%
545 }
546 \FB@addto{noextras}{%
547     \bbl@deactivate{:}\bbl@deactivate{;}%
548     \bbl@deactivate{!}\bbl@deactivate{?}%
549 }
550 \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.

```

551 \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 L<sup>A</sup>T<sub>E</sub>X. Set the default now for Plain (done later for L<sup>A</sup>T<sub>E</sub>X).

```

552 \def\autospace@beforeFDP{%
553     \ifFB@luatex@punct\FB@addDPspace=1 \fi
554     \def\FDP@thinspace{\penalty\@M\FBthinspace}%
555     \def\FDP@colonspace{\penalty\@M\FBcolonspace}}
556 \def\noautospace@beforeFDP{%
557     \ifFB@luatex@punct\FB@addDPspace=0 \fi

```

```

558         \let\FDP@thinspace\@empty
559         \let\FDP@colonspace\@empty}
560 \ifLaTeXe
561     \def\AutoSpaceBeforeFDP{\autospace@beforeFDP
562                               \FBAutoSpacePunctuationtrue}
563     \def\NoAutoSpaceBeforeFDP{\noautospace@beforeFDP
564                               \FBAutoSpacePunctuationfalse}
565     \AtEndOfPackage{\AutoSpaceBeforeFDP}
566 \else
567     \let\AutoSpaceBeforeFDP\autospace@beforeFDP
568     \let\NoAutoSpaceBeforeFDP\noautospace@beforeFDP
569     \AutoSpaceBeforeFDP
570 \fi

```

In  $\text{\LaTeX}_2\epsilon$  `\ttfamily` (and hence `\texttt`) will be redefined ‘AtBeginDocument’ as `\ttfamilyFB` so that no space is added before the four ; : ! ? characters, even if `AutoSpacePunctuation` is `true`. `\rmfamily` and `\sffamily` need to be redefined also (`\ttfamily` is not always used inside a group, its effect can be cancelled by `\rmfamily` or `\sffamily`).

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

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

```

571 \ifLaTeXe
572     \DeclareRobustCommand\ttfamilyFB{%
573         \FBAutoSpaceGuillfalse
574         \ifFB@luatex@punct\FB@addGUILspace=0 \fi
575         \noautospace@beforeFDP\ttfamilyORI}%
576     \DeclareRobustCommand\rmfamilyFB{%
577         \FBAutoSpaceGuilltrue
578         \ifFB@luatex@punct\FB@addGUILspace=1 \fi
579         \ifFBAutoSpacePunctuation
580             \autospace@beforeFDP
581         \else
582             \noautospace@beforeFDP
583         \fi
584         \rmfamilyORI}%
585     \DeclareRobustCommand\sffamilyFB{%
586         \FBAutoSpaceGuilltrue
587         \ifFB@luatex@punct\FB@addGUILspace=1 \fi
588         \ifFBAutoSpacePunctuation
589             \autospace@beforeFDP
590         \else
591             \noautospace@beforeFDP
592         \fi
593         \sffamilyORI}%
594 \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.

```
595 \newcommand*{\NoAutoSpacing}{\FBAutoSpaceGuillfalse
596   \ifFB@active@punct\shorthandoff{;:!?}\fi
597   \ifFB@xetex@punct\XeTeXinterchartokenstate=0 \fi
598   \ifFB@luatex@punct\FB@addDPspace=0 \FB@addGUILspace=0 \fi
599 }
```

## 2.3 Commands for French quotation marks

**\og** The top macros for quotation marks will be called **\og** (“ouvrez guillemets”) and

**\fg** **\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**.

```
600 \newcommand*{\og}{\@empty}
601 \newcommand*{\fg}{\@empty}
```

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

```
602 \ifLaTeXe
603 \else
604   \ifFBunicode
605     \def\guillemotleft{{\char"00AB}}
606     \def\guillemotright{{\char"00BB}}
607     \def\textquotedblleft{{\char"201C}}
608     \def\textquotedblright{{\char"201D}}
609   \else
610     \def\guillemotleft{\leavevmode\raise0.25ex
611       \hbox{$\scriptscriptstyle\ll$}}
612     \def\guillemotright{\raise0.25ex
613       \hbox{$\scriptscriptstyle\gg$}}
614     \def\textquotedblleft{''}
615     \def\textquotedblright{''}
616   \fi
617   \let\xspace\relax
618 \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.

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

`\FBguillspace` is not used with LuaTeX.

```
624 \ifFB@luatex@punct
625   \DeclareRobustCommand*{\FB@og}{\leavevmode
626       \bgroup\FB@addGUILLspace=1 \guillemotleft\egroup}
627   \DeclareRobustCommand*{\FB@fg}{\ifdim\lastskip>\z@\unskip\fi
628       \bgroup\FB@addGUILLspace=1 \guillemotright\egroup\xspace}
629 \fi
```

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

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

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

```
646 \ifLaTeXe
647   \def\bbl@frenchguillemets{\renewcommand*{\og}{\FB@og}%
648       \renewcommand*{\fg}{\FB@fg}}
649   \renewcommand*{\og}{\textquotedblleft}
650   \renewcommand*{\fg}{\ifdim\lastskip>\z@\unskip\fi \textquotedblright}
651 \else
```

```

652 \def\bbl@frenchguillemets{\let\og\FB@og
653 \let\fg\FB@fg}
654 \def\og{\textquotedblleft}
655 \def\fg{\ifdim\lastskip>\z@ \unskip\fi \textquotedblright}
656 \fi
657 \FB@addto{extras}{\babel@save\og \babel@save\fg \bbl@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...

```

658 \newcommand*\ogi{\FB@og}
659 \newcommand*\fgi{\FB@fg}
660 \newcommand*\ogii{\textquotedblleft}
661 \newcommand*\fgii{\textquotedblright}
662 \newcount\FBguill@level
663 \newif\ifFBcloseguill \FBcloseguilltrue
664 \newif\ifFBInnerGuillSingle
665 \def\FBguillopen{\bgroup\NoAutoSpacing\guillemotleft\egroup}
666 \def\FBguillclose{\bgroup\NoAutoSpacing\guillemotright\egroup}
667 \let\FBguillnone\relax
668 \let\FBeveryparguill\FBguillopen
669 \ifFB@luatex@punct
670 \let\FBeverylineguill\FBguillopen
671 \ifLaTeXe
672 \let\local@leftbox\luatexlocalleftbox
673 \else
674 \let\local@leftbox\lcalleftbox
675 \fi
676 \else
677 \let\FBeverylineguill\FBguillnone
678 \fi

```

The main command \frquote accepts (in L<sup>A</sup>T<sub>E</sub>X<sub>2<sub>ε</sub></sub> 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.

```

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

```

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

```

686 \newcommand{\fr@quote}[1]{%
687 \bgroup
688 \ifnum\FBguill@level=2
689 \PackageWarning{frenchb.ldf}{%
690 \protect\frquote\space accepts no more than two levels

```



```

691             \MessageBreak of quotations. Reported}
692     \else
693       \advance\FBguill@level by \@ne
694     \fi
695     \ifnum\FBguill@level=1

```

Set \FBeverypar@quote for level 1 quotations:

```

696     \ifx\FBeveryparguill\relax
697       \let\FBeverypar@quote\relax
698     \else
699       \def\FBeverypar@quote{\FBeveryparguill
700                               \kern.8\fontdimen2\font}%
701     \fi
702     \expandafter\csname og\romannumeral\FBguill@level\endcsname
703     \everypar \expandafter{\the\everypar \FBeverypar@quote}%
704     #1%
705     \expandafter\csname fg\romannumeral\FBguill@level\endcsname
706   \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.

```

707     \ifx\FBverylineguill\FBguillopen
708       \local@leftbox{\guillemotleft\kern.8\fontdimen2\font}%
709       \let\FBeverypar@quote\relax
710       \FB@og #1\ifFBcloseguill\FB@fg\fi
711     \else
712       \ifx\FBverylineguill\FBguillclose
713         \local@leftbox{\guillemotright\kern.8\fontdimen2\font}%
714         \let\FBeverypar@quote\relax
715         \FB@og #1\ifFBcloseguill\FB@fg\fi
716       \else

```

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

```

717       \let\FBeverypar@quote\relax
718       \ifFBInnerGuillSingle
719         \def\ogii{\leavevmode
720                 \guilsinglleft\FBguillspace}%
721         \def\fgii{\ifdim\lastskip>\z@\unskip\fi
722                 \FBguillspace\guilsinglright}%
723         \ifx\FBeveryparguill\FBguillopen
724           \def\FBeverypar@quote{\guilsinglleft
725                                   \kern.8\fontdimen2\font}%
726         \fi
727         \ifx\FBeveryparguill\FBguillclose
728           \def\FBeverypar@quote{\guilsinglright
729                                   \kern.8\fontdimen2\font}%
730         \fi
731       \fi
732       \expandafter\csname og\romannumeral\FBguill@level\endcsname
733       #1%

```

```

734         \ifFBcloseguill\expandafter
735         \csname fg\romannumeral\FBguill@level\endcsname
736     \fi
737 \fi
738 \fi
739 \fi
740 \egroup
741 }

```

## 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.

```

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

```

## 2.5 Extra utilities

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

`\up` \up eases the typesetting of superscripts like '1<sup>er</sup>'. Up to version 2.0 of frenchb  
`\fup` \fup 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).

```
777 \newif\ifFB@poorman
778 \newdimen\FB@Mht
779 \ifLaTeXe
780 \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{}`.

```
781 \newcommand*{\FBsupR}{-0.12}
782 \newcommand*{\FBsupS}{0.65}
783 \newcommand*{\FB@lc}[1]{\MakeLowercase{#1}}
784 \DeclareRobustCommand*{\FB@up@fake}[1]{%
785   \settoheight{\FB@Mht}{M}%
786   \addtolength{\FB@Mht}{\FBsupR \FB@Mht}%
787   \addtolength{\FB@Mht}{-\FBsupS ex}%
788   \raisebox{\FB@Mht}{\scalefont{\FBsupS}{\FB@lc{#1}}}%
789 }
```

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.

```

790 \def\FB@split#1#2#3#4\@nil{\def\FB@firstthree{#1#2#3}%
791                               \def\FB@suffix{#4}}
792 \def\FB@x{x}
793 \def\FB@j{j}
794 \DeclareRobustCommand*\FB@up[1]{%
795   \bgroup \FB@poormantrue
796   \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.

```

797   \edef\reserved@a{\lowercase{%
798     \noexpand\IfFileExists{\fb@encoding\FB@firstthree -sup.fd}}}%
799   \reserved@a
800   {\ifx\FB@suffix\FB@x \FB@poormanfalse\fi
801    \ifx\FB@suffix\FB@j \FB@poormanfalse\fi
802    \ifFB@poorman \FB@up@fake{#1}%
803    \else        \FB@up@real{#1}%
804    \fi}%
805   {\FB@up@fake{#1}}%
806   \egroup}

```

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

```

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

```

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

```

809 \DeclareRobustCommand*\fup[1]{%
810   \ifx\realsuperscript\@undefined
811     \FB@up{#1}%
812   \else
813     \bgroup\let\fakesuperscript\FB@up@fake
814     \realsuperscript{\FB@lc{#1}}\egroup
815   \fi}

```

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

```

816 \providecommand*\up{\relax}

```

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

```

817 \else
818 \providecommand*\up[1]{\leavevmode\raise1ex\hbox{\sevenrm #1}}
819 \fi

```

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

```

\ier 820 \def\ieme{\up{\lowercase{e}}\xspace}
\iere 821 \def\ieres{\up{\lowercase{es}}\xspace}
\ienes 822 \def\ier{\up{\lowercase{er}}\xspace}
\iers 823 \def\iers{\up{\lowercase{ers}}\xspace}
\ieres

```

```

824 \def\iere{\up{\lowercase{re}}\xspace}
825 \def\ieres{\up{\lowercase{res}}\xspace}

```

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

```

\no 826 \newcommand*{\FrenchEnumerate}[1]{%
\nos 827      #1\up{\lowercase{o}}\kern+.3em}
\nos 828 \newcommand*{\FrenchPopularEnumerate}[1]{%
\primo 829      #1\up{\lowercase{o}}\kern+.3em}
\frprimo) Typing \primo should result in ‘1°’,

```

```

830 \def\primo{\FrenchEnumerate1}
831 \def\secundo{\FrenchEnumerate2}
832 \def\tertio{\FrenchEnumerate3}
833 \def\quarto{\FrenchEnumerate4}

```

while typing `\frprimo` gives ‘1°’.

```

834 \def\frprimo{\FrenchPopularEnumerate1}
835 \def\frsecundo{\FrenchPopularEnumerate2}
836 \def\frtertio{\FrenchPopularEnumerate3}
837 \def\frquarto{\FrenchPopularEnumerate4}

```

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

```

838 \DeclareRobustCommand*{\No}{N\up{\lowercase{o}}\kern+.2em}
839 \DeclareRobustCommand*{\no}{n\up{\lowercase{o}}\kern+.2em}
840 \DeclareRobustCommand*{\Nos}{N\up{\lowercase{os}}\kern+.2em}
841 \DeclareRobustCommand*{\nos}{n\up{\lowercase{os}}\kern+.2em}

```

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

```

842 \DeclareRobustCommand*{\bsc}[1]{\leavevmode\beginngroup\kern0pt
843      \scshape #1\endgroup}
844 \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.

```

845 \ifFBunicode
846   \newcommand*{\at}{{\char"0040}}
847   \newcommand*{\circonflexe}{{\char"005E}}
848   \newcommand*{\tild}{{\char"007E}}
849   \newcommand*{\boi}{{\textbackslash}}
850   \newcommand*{\degree}{{\char"00B0}}
851 \else
852   \ifLaTeXe
853     \DeclareTextSymbol{\at}{T1}{64}

```

```

854 \DeclareTextSymbol{\circonflexe}{T1}{94}
855 \DeclareTextSymbol{\tild}{T1}{126}
856 \DeclareTextSymbolDefault{\at}{T1}
857 \DeclareTextSymbolDefault{\circonflexe}{T1}
858 \DeclareTextSymbolDefault{\tild}{T1}
859 \DeclareRobustCommand*\boi{\textbackslash}
860 \DeclareRobustCommand*\degree{\r{}}
861 \else
862 \def\T@one{T1}
863 \ifx\f@encoding\T@one
864 \newcommand*\degree{{\char6}}
865 \else
866 \newcommand*\degree{{\char23}}
867 \fi
868 \newcommand*\at{{\char64}}
869 \newcommand*\circonflexe{{\char94}}
870 \newcommand*\tild{{\char126}}
871 \newcommand*\boi{{\backslash}}
872 \fi
873 \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.

```

874 \ifLaTeXe
875 \newcommand*\degrees{\degree}
876 \ifFBunicode
877 \DeclareRobustCommand*\degrees{\degree}
878 \else
879 \def\Warning@degree@TSone{%
880 \PackageWarning{frenchb.ldf}{%
881 Degrees would look better in TS1-encoding:%
882 \MessageBreak add \protect
883 \usepackage{textcomp} to the preamble.%
884 \MessageBreak Degrees used}}
885 \AtBeginDocument{\ifx\DeclareEncodingSubset\undefined
886 \DeclareRobustCommand*\degrees{%
887 \leavevmode\hbox to 0.3em{\hss\degree\hss}%
888 \Warning@degree@TSone
889 \global\let\Warning@degree@TSone\relax}%
890 \else
891 \DeclareRobustCommand*\degrees{%
892 \hbox{\UseTextSymbol{TS1}{\textdegree}}}%
893 \fi
894 }

```

```

895 \fi
896 \else
897 \newcommand*{\degrees}{%
898 \leavevmode\hbox to 0.3em{\hss\degre\hss}}
899 \fi

```

## 2.6 Formatting numbers

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

```

900 \newcount\std@mcc
901 \newcount\dec@mcc
902 \std@mcc=\mathcode'\,
903 \dec@mcc=\std@mcc
904 \@tempcnta=\std@mcc
905 \divide\@tempcnta by "1000
906 \multiply\@tempcnta by "1000
907 \advance\dec@mcc by -\@tempcnta
908 \newcommand*{\DecimalMathComma}{\iflanguage{french}%
909 \mathcode'\,=\dec@mcc}%
910 \FB@addto{extras}{\mathcode'\,=\dec@mcc}%
911 }
912 \newcommand*{\StandardMathComma}{\mathcode'\,=\std@mcc
913 \FB@addto{extras}{\mathcode'\,=\std@mcc}%
914 }
915 \FB@addto{noextras}{\mathcode'\,=\std@mcc}

```

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

```

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

```

The next definitions only make sense for  $\text{\LaTeX} 2_{\epsilon}$ . 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.

```

918 \let\FBstop@here\relax
919 \def\FBclean@on@exit{\let\ifLaTeXe\undefined
920 \let\LaTeXettrue\undefined
921 \let\LaTeXefalse\undefined}
922 \ifx\magnification\@undefined
923 \else

```

```

924 \def\FBstop@here{\ifFB@luatex@punct
925                 \activate@luatexpunct
926                 \fi
927                 \FBclean@on@exit
928                 \ldf@quit\CurrentOption\endinput}
929 \fi
930 \FBstop@here

```

What follows is for L<sup>A</sup>T<sub>E</sub>X<sub>2 $\epsilon$</sub>  *only*; as all L<sup>A</sup>T<sub>E</sub>X<sub>2 $\epsilon$</sub>  based formats include  $\epsilon$ -T<sub>E</sub>X, we can use `\ifdefined` now. We redefine `\nombre` for L<sup>A</sup>T<sub>E</sub>X<sub>2 $\epsilon$</sub> . 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.

```

931 \renewcommand*{\nombre}[1]{\Warning@nombre\numprint{#1}}
932 \newcommand*{\Warning@nombre}{%
933   \ifdefined\numprint
934   \else
935     \PackageWarning{frenchb.ldf}{%
936       \protect\nombre\space now relies on package numprint.sty,%
937       \MessageBreak add \protect
938       \usepackage[autolanguage]{numprint}\MessageBreak
939       to your preamble *after* loading babel,\MessageBreak
940       see file numprint.pdf for more options.\MessageBreak
941       \protect\nombre\space called}%
942   \global\let\Warning@nombre\relax
943   \fi
944 }

```

## 2.7 Caption names

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

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

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

```

945 \StartBabelCommands*{\BabelLanguages}{captions}
946   [unicode, fontenc=EU1 EU2, charset=utf8]
947   \SetString{\refname}{Références}
948   \SetString{\abstractname}{Résumé}
949   \SetString{\prefacename}{Préface}
950   \SetString{\contentsname}{Table des matières}
951   \SetString{\ccname}{Copie à }
952   \SetString{\proofname}{Démonstration}
953   \SetStringLoop{ordinal#1}{%
954     Première,Deuxième,Troisième,Quatrième,Cinquième,%
955     Sixième,Septième,Huitième,Neuvième,Dixième,Onzième,%
956     Douzième,Treizième,Quatorzième,Quinzième,Seizième,%
957     Dix-septième,Dix-huitième,Dix-neuvième,Vingtième}
958 \StartBabelCommands*{\BabelLanguages}{captions}

```



```

959 \SetString{\refname}{R\ref\erences}
960 \SetString{\abstractname}{R\esum\e}
961 \SetString{\bibname}{Bibliographie}
962 \SetString{\prefacename}{Pr\eface}
963 \SetString{\chaptername}{Chapitre}
964 \SetString{\appendixname}{Annexe}
965 \SetString{\contentsname}{Table des mati\eres}
966 \SetString{\listfigurename}{Table des figures}
967 \SetString{\listtablename}{Liste des tableaux}
968 \SetString{\indexname}{Index}
969 \SetString{\figurename}{\scshape Figure}
970 \SetString{\tablename}{\scshape Table}
971 \SetString{\pagename}{page}
972 \SetString{\seename}{voir}
973 \SetString{\alsoname}{voir aussi}
974 \SetString{\enclname}{P.~J. }
975 \SetString{\ccname}{Copie \a }
976 \SetString{\headtoname}{}
977 \SetString{\proofname}{D\emonstration}
978 \SetString{\glossaryname}{Glossaire}

“Première partie” instead of “Part I”.

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

```

Up to v2.6h frenchb used to merge `\captionsfrenchb` and `\captionsfrancais` 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  $\text{\LaTeX}2_{\epsilon}$  classes; the ‘:’ is made active too late, no space is added before it. With Lua $\text{\LaTeX}$  and Xe $\text{\LaTeX}$ , this glitch doesn’t occur, you get ‘Figure 1 :’ which is correct in French. With pdf $\text{\LaTeX}$  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  $\text{\LaTeX}2_{\epsilon}$  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.lda` makes `' >` active).

```

992 \bgroup
993   \catcode'=:12 \catcode'>=12 \relax
994   \long\gdef\STD@makecaption#1#2{%
995     \vskip\abovecaptionskip
996     \sbox\@tempboxa{#1: #2}%
997     \ifdim \wd\@tempboxa >\hsize
998       #1: #2\par
999     \else
1000       \global \@minipagefalse
1001       \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
1002       \fi
1003       \vskip\belowcaptionskip}
1004 \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.

```

1005 \newif\if@FBwarning@capsep
1006 \@FBwarning@capseptrue
1007 \newcommand{\FBWarning}[2]{\PackageWarning{#1}{#2}}
1008 \newcommand*\CaptionSeparator{\space\textendash\space}
1009 \def\FBCaption@Separator{:\space}
1010 \long\def\FB@makecaption#1#2{%
1011   \vskip\abovecaptionskip
1012   \sbox\@tempboxa{#1\FBCaption@Separator #2}%
1013   \ifdim \wd\@tempboxa >\hsize
1014     #1\FBCaption@Separator #2\par
1015   \else
1016     \global \@minipagefalse
1017     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
1018     \fi
1019     \vskip\belowcaptionskip}

```

Disable the standard warning with `AMS` and `SMF` classes.

```

1020 \ifclassloaded{amsart}{\@FBwarning@capsepfalse}{}
1021 \ifclassloaded{amsbook}{\@FBwarning@capsepfalse}{}
1022 \ifclassloaded{amsdtx}{\@FBwarning@capsepfalse}{}

```

```

1023 \@ifclassloaded{amslatex}{\@FBwarning@capsepfalse}{}
1024 \@ifclassloaded{amproc}{\@FBwarning@capsepfalse}{}
1025 \@ifclassloaded{smfart}{\@FBwarning@capsepfalse}{}
1026 \@ifclassloaded{smfbook}{\@FBwarning@capsepfalse}{}

```

Disable the standard warning unless high punctuation is active.

```

1027 \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). No warning either if \@makecaption is undefined.

```

1028 \newif\ifFB@koma
1029 \@ifclassloaded{memoir}{\@FBwarning@capsepfalse}{}
1030 \@ifclassloaded{scrartcl}{\@FBwarning@capsepfalse\FB@komatrue}{}
1031 \@ifclassloaded{scrbook}{\@FBwarning@capsepfalse\FB@komatrue}{}
1032 \@ifclassloaded{scrreprt}{\@FBwarning@capsepfalse\FB@komatrue}{}
1033 \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.

```

1034 \@ifpackageloaded{caption}
1035   {\FBWarning{frenchb.lfd}%
1036     {Please load the "caption" package\MessageBreak
1037       AFTER babel/frenchb; reported}%
1038     \@FBwarning@capsepfalse}%
1039   {}

```

Same for package floatrow.

```

1040 \@ifpackageloaded{floatrow}
1041   {\FBWarning{frenchb.lfd}%
1042     {Please load the "floatrow" package\MessageBreak
1043       AFTER babel/frenchb; reported}%
1044     \@FBwarning@capsepfalse}%
1045   {}

```

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').

```

1046 \AtBeginDocument{%
1047   \ifx\@makecaption\STD@makecaption
1048     \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.

```

1049   \ifFBOldFigTabCaptions
1050   \else
1051     \def\FBCaption@Separator{\iflanguage{french}{ : }{: }}%
1052   \fi
1053   \ifFBCustomiseFigTabCaptions
1054     \ifx\bbbl@main@language\FB@french

```

```

1055         \def\FBCaption@Separator{\CaptionSeparator}%
1056     \fi
1057 \fi
1058 \if@FBwarning@capsepfalse
1059 \fi
1060 \if@FBwarning@capsep
1061     \FBWarning{frenchb.ldb}%
1062     {Figures' and tables' captions might look like\MessageBreak
1063     'Figure 1:' which is wrong in French.\MessageBreak
1064     Check your class or packages to change this;\MessageBreak
1065     reported}%
1066 \fi
1067 \let\FB@makecaption\relax
1068 \let\STD@makecaption\relax
1069 }

```

## 2.8 Dots...

**\FBtextellipsis** L<sup>A</sup>T<sub>E</sub>X<sub>2<sub>ε</sub></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 L<sup>A</sup>T<sub>E</sub>X<sub>2<sub>ε</sub></sub> only).

The `\if` construction in the L<sup>A</sup>T<sub>E</sub>X<sub>2<sub>ε</sub></sub> definition of `\dots` doesn't allow the use of `xspace` (`xspace` is always followed by a `\fi`), so we use the AMS-L<sup>A</sup>T<sub>E</sub>X 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.

```

1070 \ifFBunicode
1071     \let\FBtextellipsis\textellipsis
1072 \else
1073     \DeclareTextSymbol{\FBtextellipsis}{LY1}{133}
1074     \DeclareTextCommandDefault{\FBtextellipsis}{%
1075         .\kern\fontdimen3\font.\kern\fontdimen3\font.\xspace}
1076 \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 L<sup>A</sup>T<sub>E</sub>X 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.

```

1077 \newcommand*{\Tdots@}{\@xp\textellipsis}
1078 \newcommand*{\Mdots@}{\@xp\mdots@}
1079 \AtBeginDocument{\DeclareRobustCommand*\dots}{\relax
1080     \csname\ifmode M\else T\fi dots@\endcsname}%
1081     \ifdefined\@xp\else\let\@xp\relax\fi
1082     \ifdefined\mdots@\else\let\Mdots@\mathellipsis\fi
1083 }
1084 \def\bbl@frenchdots{\babel@save\Tdots@ \let\Tdots@\FBtextellipsis}
1085 \FB@addto{extras}{\bbl@frenchdots}

```

## 2.9 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.

```
1086 \newcommand*{\frenchbsetup}[1]{%
1087   \setkeys{FB}{#1}%
1088 }%
1089 \@onlypreamble\frenchbsetup
```

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

```
1090 \newif\ifFBShowOptions           \FBShowOptionsfalse
1091 \newif\ifFBStandardLayout        \FBStandardLayouttrue
1092 \newif\ifFBGlobalLayoutFrench    \FBGlobalLayoutFrenchtrue
1093 \newif\ifFBReduceListSpacing     \FBReduceListSpacingfalse
1094 \newif\ifFBListOldLayout         \FBListOldLayoutfalse
1095 \newif\ifFBCompactItemize        \FBCompactItemizefalse
1096 \newif\ifFBStandardItemizeEnv    \FBStandardItemizeEnvtrue
1097 \newif\ifFBStandardEnumerateEnv  \FBStandardEnumerateEnvtrue
1098 \newif\ifFBStandardItemLabels    \FBStandardItemLabelstrue
1099 \newif\ifFBStandardLists         \FBStandardListstrue
1100 \newif\ifFBIndentFirst           \FBIndentFirstfalse
1101 \newif\ifFBFrenchFootnotes       \FBFrenchFootnotesfalse
1102 \newif\ifFBAutoSpaceFootnotes    \FBAutoSpaceFootnotesfalse
1103 \newif\ifFBOriginalTypewriter    \FBOriginalTypewriterfalse
1104 \newif\ifFBThinColonSpace        \FBThinColonSpacefalse
1105 \newif\ifFBThinSpaceInFrenchNumbers \FBThinSpaceInFrenchNumbersfalse
1106 \newif\ifFBFrenchSuperscripts    \FBFrenchSuperscriptstrue
1107 \newif\ifFBLowercaseSuperscripts \FBLowercaseSuperscriptstrue
1108 \newif\ifFBPartNameFull          \FBPartNameFulltrue
1109 \newif\ifFBCustomiseFigTabCaptions \FBCustomiseFigTabCaptionsfalse
1110 \newif\ifFBOldFigTabCaptions    \FBOldFigTabCaptionsfalse
1111 \newif\ifFBSuppressWarning       \FBSuppressWarningfalse
1112 \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.

```

1113 \edef\FB@french{\CurrentOption}
1114 \AtEndOfPackage{%
1115   \ifx\bbl@main@language\FB@french
1116     \FBGlobalLayoutFrenchtrue
1117     \@ifclassloaded{beamer}%
1118       {\PackageInfo{frenchb.ldf}{%
1119         No list customisation for the beamer class,%
1120         \MessageBreak reported}}%
1121       {\FBReduceListSpacingtrue
1122        \FBStandardItemizeEnvfalse
1123        \FBStandardEnumerateEnvfalse
1124        \FBStandardItemLabelsfalse}%
1125     \FBIndentFirsttrue
1126     \FBFrenchFootnotesttrue
1127     \FBAutoSpaceFootnotesttrue
1128     \FBCustomiseFigTabCaptionstrue
1129   \else
1130     \FBGlobalLayoutFrenchfalse
1131   \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.

```

1132 \RequirePackage{keyval}%
1133 \define@key{FB}{ShowOptions}[true]%
1134   {\csname FBShowOptions#1\endcsname}%
1135 \define@key{FB}{StandardLayout}[true]%
1136   {\csname FBStandardLayout#1\endcsname
1137   \ifFBStandardLayout
1138     \FBReduceListSpacingfalse
1139     \FBStandardItemizeEnvtrue
1140     \FBStandardItemLabelstrue
1141     \FBStandardEnumerateEnvtrue
1142     \FBIndentFirstfalse
1143     \FBFrenchFootnotesfalse
1144     \FBAutoSpaceFootnotesfalse
1145     \FBGlobalLayoutFrenchfalse
1146   \else
1147     \FBReduceListSpacingtrue
1148     \FBStandardItemizeEnvfalse
1149     \FBStandardItemLabelsfalse
1150     \FBStandardEnumerateEnvfalse
1151     \FBIndentFirsttrue
1152     \FBFrenchFootnotesttrue

```

```

1153             \FBAutoSpaceFootnotesttrue
1154             \fi}%
1155 \define@key{FB}{GlobalLayoutFrench}[true]%
1156             {\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.

1157             \ifFBGlobalLayoutFrench
1158             \ifx\bbbl@main@language\FB@french
1159             \else
1160             \PackageWarning{frenchb.ldf}%
1161             {Option 'GlobalLayoutFrench' skipped:%
1162             \MessageBreak French is *not*
1163             babel's last option.\MessageBreak}%
1164             \fi
1165             \fi}%
1166 \define@key{FB}{ReduceListSpacing}[true]%
1167             {\csname FBReduceListSpacing#1\endcsname}%
1168 \define@key{FB}{ListOldLayout}[true]%
1169             {\csname FBListOldLayout#1\endcsname
1170             \ifFBListOldLayout
1171             \FBStandardEnumerateEnvtrue
1172             \renewcommand*\FrenchLabelItem{\textendash}%
1173             \fi}%
1174 \define@key{FB}{CompactItemize}[true]%
1175             {\csname FBCompactItemize#1\endcsname
1176             \ifFBCompactItemize
1177             \FBStandardItemizeEnvfalse
1178             \FBStandardEnumerateEnvfalse
1179             \else
1180             \FBStandardItemizeEnvtrue
1181             \FBStandardEnumerateEnvtrue
1182             \fi}%
1183 \define@key{FB}{StandardItemizeEnv}[true]%
1184             {\csname FBStandardItemizeEnv#1\endcsname}%
1185 \define@key{FB}{StandardEnumerateEnv}[true]%
1186             {\csname FBStandardEnumerateEnv#1\endcsname}%
1187 \define@key{FB}{StandardItemLabels}[true]%
1188             {\csname FBStandardItemLabels#1\endcsname}%
1189 \define@key{FB}{ItemLabels}{%
1190     \renewcommand*\FrenchLabelItem{#1}}%
1191 \define@key{FB}{ItemLabeli}{%
1192     \renewcommand*\Frlabelitemi{#1}}%
1193 \define@key{FB}{ItemLabelii}{%
1194     \renewcommand*\Frlabelitemii{#1}}%
1195 \define@key{FB}{ItemLabeliii}{%
1196     \renewcommand*\Frlabelitemiii{#1}}%
1197 \define@key{FB}{ItemLabeliv}{%
1198     \renewcommand*\Frlabelitemiv{#1}}%
1199 \define@key{FB}{StandardLists}[true]%

```

```

1200             {\csname FBStandardLists#1\endcsname
1201             \ifFBStandardLists
1202                 \FBReduceListSpacingfalse
1203                 \FBCompactItemizefalse
1204                 \FBStandardItemizeEnvtrue
1205                 \FBStandardEnumerateEnvtrue
1206                 \FBStandardItemLabelstrue
1207             \else
1208                 \FBReduceListSpacingtrue
1209                 \FBCompactItemizetrue
1210                 \FBStandardItemizeEnvfalse
1211                 \FBStandardEnumerateEnvfalse
1212                 \FBStandardItemLabelsfalse
1213             \fi}%
1214 \define@key{FB}{IndentFirst}[true]%
1215     {\csname FBIndentFirst#1\endcsname}%
1216 \define@key{FB}{FrenchFootnotes}[true]%
1217     {\csname FBFrenchFootnotes#1\endcsname}%
1218 \define@key{FB}{AutoSpaceFootnotes}[true]%
1219     {\csname FBAutoSpaceFootnotes#1\endcsname}%
1220 \define@key{FB}{AutoSpacePunctuation}[true]%
1221     {\csname FBAutoSpacePunctuation#1\endcsname}%
1222 \define@key{FB}{OriginalTypewriter}[true]%
1223     {\csname FBOriginalTypewriter#1\endcsname}%
1224 \define@key{FB}{ThinColonSpace}[true]%
1225     {\csname FBThinColonSpace#1\endcsname}%
1226 \define@key{FB}{ThinSpaceInFrenchNumbers}[true]%
1227     {\csname FBThinSpaceInFrenchNumbers#1\endcsname}%
1228 \define@key{FB}{FrenchSuperscripts}[true]%
1229     {\csname FBFrenchSuperscripts#1\endcsname}%
1230 \define@key{FB}{LowercaseSuperscripts}[true]%
1231     {\csname FBLowercaseSuperscripts#1\endcsname}%
1232 \define@key{FB}{PartNameFull}[true]%
1233     {\csname FBPartNameFull#1\endcsname
1234     \ifFBPartNameFull
1235     \else
1236         \FB@addto{captions}{%
1237             \def\partname{Partie\protect\FB@Rpart}}}%
1238     \fi}%
1239 \define@key{FB}{CustomiseFigTabCaptions}[true]%
1240     {\csname FBCustomiseFigTabCaptions#1\endcsname}%
1241 \define@key{FB}{OldFigTabCaptions}[true]%
1242     {\csname FBOldFigTabCaptions#1\endcsname
    \CurrentOption no longer defined. It's value has been saved in \FB@CurOpt while
    reading frenchb. ldf.
1243     \ifFBOldFigTabCaptions
1244         \FB@addto{extras}{\babel@save\FBCaption@Separator
1245             \def\FBCaption@Separator{\CaptionSeparator}}}%
1246     \fi}%
1247 \define@key{FB}{SuppressWarning}[true]%

```



```

1248         {\csname FBSuppressWarning#1\endcsname
1249         \ifFBSuppressWarning
1250         \renewcommand{\FBWarning}[2]{\relax}%
1251         \fi}%

```

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

```

1252 \define@key{FB}{INGuillSpace}[true]%
1253         {\csname FBINGuillSpace#1\endcsname}%
1254 \define@key{FB}{InnerGuillSingle}[true]%
1255         {\csname FBInnerGuillSingle#1\endcsname}%
1256 \define@key{FB}{EveryParGuill}{\expandafter\let\expandafter
1257         \FBeveryparguill\csname FBguill#1\endcsname}%
1258 \define@key{FB}{EveryLineGuill}{\expandafter\let\expandafter
1259         \FBeverylineguill\csname FBguill#1\endcsname
1260         \ifFB@luatex@punct
1261         \else
1262         \let\FBeverylineguill\FBguillnone
1263         \PackageWarning{frenchb.ldb}%
1264             {Option ‘EveryLineGuill’ skipped:%
1265             \MessageBreak this option is for
1266             LuaTeX *only*.\MessageBreak Reported}%
1267         \fi}%

```

Inputting French quotes as *single characters* when they are available on the keyboard (through a compose key for instance) is more comfortable than typing `\og` and `\fg`.

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@addGUILLspace` attribute for LuaTeX or set `\XeTeXcharclass` of quotes to the proper value for XeTeX.

```

1268 \define@key{FB}{og}{%
1269     \ifFBunicode

```

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

```

1270         \ifFB@luatex@punct
1271         \FB@addGUILLspace=1 \relax
1272         \fi

```

then with XeTeX it is a bit more tricky:

```

1273         \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).

```

1274         \XeTeXcharclass"13 = \FB@guilo
1275         \XeTeXcharclass"AB = \FB@guilo
1276         \XeTeXcharclass"A0 = \FB@guilnul
1277         \XeTeXcharclass"202F = \FB@guilnul
1278     \fi
1279 \else

```

This is for conventional TeX engines:

```

1280     \newcommand*\FB@@og{%
1281         \iflanguage{french}%
1282             {\ifFBAutoSpaceGuill\FB@og\ignorespaces
1283              \else\guillemotleft
1284              \fi}%
1285             {\guillemotleft}}%
1286     \AtBeginDocument{%
1287         \ifdefined\DeclareInputText
1288         \ifdefined\uc@dclc

```

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

```

1289         \uc@dclc{171}{default}{\FB@@og}%
1290     \else

```

if encoding is not utf8x, try utf8...

```

1291         \ifdefined\DeclareUnicodeCharacter

```

utf8 loaded, use \DeclareUnicodeCharacter,

```

1292         \DeclareUnicodeCharacter{00AB}{\FB@@og}%
1293     \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.

```

1294         \@tempcnta'1\relax
1295         \ifdefined\mule@def
1296             \mule@def{11}{\FB@@og}%
1297         \else
1298             \DeclareInputText{\the\@tempcnta}{\FB@@og}%
1299         \fi
1300     \fi
1301 \fi
1302 \else

```

Package inputenc not loaded, no way...

```

1303     \PackageWarning{frenchb.ldf}%
1304     {Option 'og' requires package inputenc.\MessageBreak}%
1305 \fi
1306 }%
1307 \fi
1308 }%

```

Same code for the closing quote.

```

1309 \define@key{FB}{fg}{%
1310     \ifFBunicode
1311         \ifFB@luatex@punct
1312             \FB@addGUILspace=1 \relax
1313         \fi
1314         \ifFB@xetex@punct
1315             \XeTeXcharclass"14 = \FB@guilf
1316             \XeTeXcharclass"BB = \FB@guilf
1317             \XeTeXcharclass"A0 = \FB@guilnul
1318             \XeTeXcharclass"202F = \FB@guilnul
1319         \fi
1320     \else
1321         \newcommand*{\FB@@fg}{%
1322             \iflanguage{french}%
1323                 {\ifFBAutoSpaceGuill\FB@fg
1324                 \else\guillemotright
1325                 \fi}%
1326             {\guillemotright}}%
1327         \AtBeginDocument{%
1328             \ifdefined\DeclareInputText
1329                 \ifdefined\uc@dcl
1330                     \uc@dcl{187}{default}{\FB@@fg}%
1331                 \else
1332                     \ifdefined\DeclareUnicodeCharacter
1333                         \DeclareUnicodeCharacter{00BB}{\FB@@fg}%
1334                     \else
1335                         \@tempcnta'#1\relax
1336                         \ifdefined\mule@def
1337                             \mule@def{27}{\FB@@fg}%
1338                         \else
1339                             \DeclareInputText{\the\@tempcnta}{\FB@@fg}%
1340                         \fi
1341                     \fi
1342                 \fi
1343             \else
1344                 \PackageWarning{frenchb.ldf}%
1345                     {Option 'fg' requires package inputenc.\MessageBreak}%
1346             \fi
1347         }%
1348     \fi
1349 }%
1350 }

```

**\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.

```

1351 \newcommand*{\FBprocess@options}{%

```

Update flags if a package customising lists has been loaded, currently: enumitem, paralist, enumerate.

```

1352 \@ifpackageloaded{enumitem}{%
1353   \ifFBStandardItemizeEnv
1354   \else
1355     \FBStandardItemizeEnvtrue
1356     \PackageInfo{frenchb.ldb}%
1357     {Setting StandardItemizeEnv=true for\MessageBreak
1358      compatibility with enumitem package,\MessageBreak}%
1359   \fi
1360   \ifFBStandardEnumerateEnv
1361   \else
1362     \FBStandardEnumerateEnvtrue
1363     \PackageInfo{frenchb.ldb}%
1364     {Setting StandardEnumerateEnv=true for\MessageBreak
1365      compatibility with enumitem package,\MessageBreak}%
1366   \fi}{}%
1367 \@ifpackageloaded{paralist}{%
1368   \ifFBStandardItemizeEnv
1369   \else
1370     \FBStandardItemizeEnvtrue
1371     \PackageInfo{frenchb.ldb}%
1372     {Setting StandardItemizeEnv=true for\MessageBreak
1373      compatibility with paralist package,\MessageBreak}%
1374   \fi
1375   \ifFBStandardEnumerateEnv
1376   \else
1377     \FBStandardEnumerateEnvtrue
1378     \PackageInfo{frenchb.ldb}%
1379     {Setting StandardEnumerateEnv=true for\MessageBreak
1380      compatibility with paralist package,\MessageBreak}%
1381   \fi}{}%
1382 \@ifpackageloaded{enumerate}{%
1383   \ifFBStandardEnumerateEnv
1384   \else
1385     \FBStandardEnumerateEnvtrue
1386     \PackageInfo{frenchb.ldb}%
1387     {Setting StandardEnumerateEnv=true for\MessageBreak
1388      compatibility with enumerate package,\MessageBreak}%
1389   \fi}{}%

```

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

```

1390 \def\FB@ufl{\update@frenchlists}
1391 \ifx\bbl@main@language\FB@french
1392   \update@frenchlists
1393 \fi

```

The layout of footnotes is handled at the \begin{document} depending on the values of flags [FrenchFootnotes](#) and [AutoSpaceFootnotes](#) (see section 2.12), 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.

```
1394 \ifFBAutoSpacePunctuation
1395     \autospace@beforeFDP
1396 \else
1397     \noautospace@beforeFDP
1398 \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.

```
1399 \ifFBOriginalTypewriter
1400 \else
1401     \let\ttfamilyORI\ttfamily
1402     \let\rmfamilyORI\rmfamily
1403     \let\sffamilyORI\sffamily
1404     \let\ttfamily\ttfamilyFB
1405     \let\rmfamily\rmfamilyFB
1406     \let\sffamily\sffamilyFB
1407 \fi
```

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

```
1408 \ifFBThinColonSpace
1409     \ifFB@luatex@punct
1410         \FBcolonskip=\FBthinspace\relax
1411     \else
1412         \renewcommand*{\FBcolonspace}{\FBthinspace}%
1413     \fi
1414 \fi
```

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

```
1415 \ifBINGuillSpace
1416     \ifFB@luatex@punct
1417         \FBguillskip=3.33pt plus 1.665pt minus 1.11pt \relax
1418     \else
1419         \renewcommand*{\FBguillspace}{\space}%
1420     \fi
1421 \fi
```

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

```
1422 \@ifpackageloaded{numprint}%
1423 {\ifnprt@autolanguage
1424     \providecommand*{\npstylefrench}{}%
1425     \ifFBThinSpaceInFrenchNumbers
1426         \renewcommand*{\npstylefrench}{%
```

```

1427         \nphousandsep{\,%
1428         \npdecimalsign{,%
1429         \npproductsign{\cdot}%
1430         \npunitseparator{\,%
1431         \npdegreeseperator{}%
1432         \nppercentseparator{\nprt@unitsep}%
1433         }%
1434     \else
1435         \renewcommand*\npstylefrench{%
1436         \nphousandsep{~}%
1437         \npdecimalsign{,%
1438         \npproductsign{\cdot}%
1439         \npunitseparator{\,%
1440         \npdegreeseperator{}%
1441         \nppercentseparator{\nprt@unitsep}%
1442         }%
1443     \fi
1444     \npaddtolanguage{french}{french}%
1445 \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.

```

1446 \ifFBFrenchSuperscripts
1447     \DeclareRobustCommand*\up*{\@ifstar{\FB@up@fake}{\fup}}%
1448 \else
1449     \DeclareRobustCommand*\up*{\@ifstar{\FB@up@fake}%
1450                                     {\textsuperscript}}%
1451 \fi

```

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

```

1452 \ifFBLowercaseSuperscripts
1453 \else
1454     \renewcommand*\FB@lc[1]{##1}%
1455 \fi

```

Use French `\CaptionSeparator` for koma-script and memoir classes unless **CustomiseFigTabCaptions** has been set to `false` (the default).

```

1456 \ifBFCustomiseFigTabCaptions
1457     \ifFB@koma
1458         \let\captionformat\CaptionSeparator
1459     \fi
1460     \@ifclassloaded{memoir}%
1461     {\captiondelim{\CaptionSeparator}}{}%
1462 \else

```

When **CustomiseFigTabCaptions** is `false`, have the colon behave properly in French.

```

1463 \ifFB@koma
1464     \def\captionformat{\iflanguage{french}{ : }{: }}%

```

```

1465 \fi
1466 \@ifclassloaded{memoir}%
1467 {\captiondelim{\iflanguage{french}{ : }{: }}{}}}%
1468 \fi

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

1469 \ifFBSShowOptions
1470 \GenericWarning{* }{%
1471 * **** List of possible options for frenchb ****\MessageBreak
1472 [Default values between brackets when frenchb is loaded *LAST*]%
1473 \MessageBreak
1474 ShowOptions=true [false]\MessageBreak
1475 StandardLayout=true [false]\MessageBreak
1476 GlobalLayoutFrench=false [true]\MessageBreak
1477 StandardLists=true [false]\MessageBreak
1478 IndentFirst=false [true]\MessageBreak
1479 ReduceListSpacing=false [true]\MessageBreak
1480 ListOldLayout=true [false]\MessageBreak
1481 StandardItemizeEnv=true [false]\MessageBreak
1482 StandardEnumerateEnv=true [false]\MessageBreak
1483 StandardItemLabels=true [false]\MessageBreak
1484 ItemLabels=\textendash, \textbullet,
1485 \protect\ding{43},... [\textendash]\MessageBreak
1486 ItemLabeli=\textendash, \textbullet,
1487 \protect\ding{43},... [\textendash]\MessageBreak
1488 ItemLabelii=\textendash, \textbullet,
1489 \protect\ding{43},... [\textendash]\MessageBreak
1490 ItemLabeliii=\textendash, \textbullet,
1491 \protect\ding{43},... [\textendash]\MessageBreak
1492 ItemLabeliv=\textendash, \textbullet,
1493 \protect\ding{43},... [\textendash]\MessageBreak
1494 FrenchFootnotes=false [true]\MessageBreak
1495 AutoSpaceFootnotes=false [true]\MessageBreak
1496 AutoSpacePunctuation=false [true]\MessageBreak
1497 OriginalTypewriter=true [false]\MessageBreak
1498 ThinColonSpace=true [false]\MessageBreak
1499 ThinSpaceInFrenchNumbers=true [false]\MessageBreak
1500 FrenchSuperscripts=false [true]\MessageBreak
1501 LowercaseSuperscripts=false [true]\MessageBreak
1502 PartNameFull=false [true]\MessageBreak
1503 SuppressWarning=true [false]\MessageBreak
1504 CustomiseFigTabCaptions=false [true]\MessageBreak
1505 OldFigTabCaptions=true [false]\MessageBreak
1506 INGuillSpace=true [false]\MessageBreak
1507 InnerGuillSingle=true [false]\MessageBreak
1508 EveryParGuill=open, close, none [open]\MessageBreak
1509 EveryLineGuill=open, close, none
1510 [open in LuaTeX, none otherwise]\MessageBreak
1511 og= <left quote character>, fg= <right quote character>%
1512 \MessageBreak
1513 *****%

```

```

1514     \MessageBreak\protect\frenchbsetup{ShowOptions}}
1515   \fi
1516 }

```

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.

```

1517 \AtBeginDocument{%
1518   \providecommand*\xspace{\relax}%
1519   \let\xspace\relax
1520   \let\up\xspace
1521   \let\deg\xspace
1522   \let\degres\xspace
1523   \def\ieme{e\xspace}%
1524   \def\iemes{es\xspace}%
1525   \def\ier{er\xspace}%
1526   \def\iers{ers\xspace}%
1527   \def\iere{re\xspace}%
1528   \def\ieres{res\xspace}%
1529   \def\FrenchEnumerate#1{#1\deg\xspace}%
1530   \def\FrenchPopularEnumerate#1{#1\deg)\xspace}%
1531   \def\No{N\deg\xspace}%
1532   \def\no{n\deg\xspace}%
1533   \def\Nos{N\deg\xspace}%
1534   \def\nos{n\deg\xspace}%
1535   \def\FB@og{\guillemotleft\xspace}%
1536   \def\FB@fg{\space\guillemotright}%
1537   \def\at{@}%
1538   \def\circonflexe{\string^}%
1539   \def\tild{\string~}%
1540   \let\bsc\textsc
1541 }%
1542 \fi

```

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

```

1545   \FBprocess@options

```

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

```

1546   \ifFB@luatexpunct
1547     \activate@luatexpunct
1548   \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!

```

1549 \ifFBunicode
1550   \ifdefined\DeclareUTFcharacter
1551   \else
1552     \ifpackageloaded{luainputenc}{}%
1553       {\PackageWarning{frenchb.ldb}%
1554        {Add \protect\usepackage{fontspec} to the\MessageBreak
1555         preamble of your document,}%
1556       }%
1557   \fi
1558 \else
1559   \begingroup \newcommand{\FBOTone}{OT1}%
1560   \ifx\encodingdefault\FBOTone
1561     \PackageWarning{frenchb.ldb}%
1562       {OT1 encoding should not be used for French.%
1563       \MessageBreak
1564       Add \protect\usepackage[T1]{fontenc} to the
1565       preamble\MessageBreak of your document,}%
1566   \fi
1567   \endgroup
1568 \fi
1569 }
```

## 2.10 French lists

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

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

```

1570 \let\listORI\list
1571 \let\endlistORI\endlist
1572 \def\FB@listVsettings{%
1573   \setlength{\itemsep}{0.4ex plus 0.2ex minus 0.2ex}%
1574   \setlength{\parsep}{0.4ex plus 0.2ex minus 0.2ex}%
1575   \setlength{\topsep}{0.8ex plus 0.4ex minus 0.4ex}%
1576   \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`.

```

1577      \@tempdima=\parskip
1578      \addtolength{\topsep}{-\@tempdima}%
1579      \addtolength{\partopsep}{\@tempdima}%
1580 }
1581 \def\listFB#1#2{\listORI{#1}{\FB@listVsettings #2}}
1582 \let\endlistFB\endlist

```

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

- The ‘•’ is never used in French itemize-lists, an emdash ‘—’ or an en-dash ‘-’ is preferred for all levels. The item label to be used in French is stored in `\FrenchLabelItem`, it defaults to ‘—’ and can be changed using `\frenchbsetup{}` (see section 2.9).
- Vertical spacing between items, before and after the list, should be *null* with *no glue* added;
- In French the labels of itemize-lists are vertically aligned as follows:

```

Text starting at 'parindent'
← Leftmargin
- first item. . .
- first second level item
- next one. . .
- second item. . .

```

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

```

\FrenchLabelItem \newcommand*{\FrenchLabelItem}{\textemdash}
\FrenchLabelItemii \newcommand*{\FrenchLabelItemii}{\FrenchLabelItem}
\FrenchLabelItemiii \newcommand*{\FrenchLabelItemiii}{\FrenchLabelItem}
\FrenchLabelItemiv \newcommand*{\FrenchLabelItemiv}{\FrenchLabelItem}
\FrenchLabelItemv \newcommand*{\FrenchLabelItemv}{\FrenchLabelItem}

```

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

```

1588 \newlength\listindentFB
1589 \setlength{\listindentFB}{-1pt}
1590 \newlength\labelwidthFB
1591 \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.

```

1592 \newlength\leftmarginFB
1593 \def\FB@listHsettings{%
1594   \leftmarginFB\labelwidthFB
1595   \advance\leftmarginFB \labelsep

```

```

1596 \leftmargini\leftmarginFB
1597 \advance\leftmargini \listindentFB
1598 \leftmarginii\leftmarginFB
1599 \leftmarginiii\leftmarginFB
1600 \leftmarginiv\leftmarginFB
1601 \leftmargin\csname leftmargin\romannumeral\the\@listdepth\endcsname
1602 }

```

`\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).

```

1603 \def\FB@itemizesettings{%
1604   \ifFBReduceListSpacing
1605     \setlength{\itemsep}{\z@}%
1606     \setlength{\parsep}{\z@}%
1607     \setlength{\topsep}{\z@}%
1608     \setlength{\partopsep}{\z@}%
1609     \@tempdima=\parskip
1610     \addtolength{\topsep}{-\@tempdima}%
1611     \addtolength{\partopsep}{-\@tempdima}%
1612   \fi
1613   \settowidth{\labelwidth}{\csname\@itemitem\endcsname}%
1614   \ifFBListOldLayout
1615     \setlength{\leftmargin}{\labelwidth}%
1616     \addtolength{\leftmargin}{\labelsep}%
1617     \addtolength{\leftmargin}{\parindent}%
1618   \else
1619     \FB@listHsettings
1620   \fi
1621 }

```

The definition of `\itemizeFB` follows the one of `\itemize` in standard L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> classes (see `ltlists.dtx`), spaces are customised by `\FB@itemizesettings`.

```

1622 \def\itemizeFB{%
1623   \ifnum \@itemdepth >\thr@@\toodeep\else
1624     \advance\@itemdepth\@ne
1625     \edef\@itemitem{labelitem\romannumeral\the\@itemdepth}%
1626     \expandafter
1627     \listORI
1628     \csname\@itemitem\endcsname
1629     \FB@itemizesettings
1630   \fi
1631 }
1632 \let\enditemizeFB\endlistORI

1633 \def\labelitemsFB{%
1634   \let\labelitemi\Frlabelitemi
1635   \let\labelitemii\Frlabelitemii
1636   \let\labelitemiii\Frlabelitemiii

```

```

1637 \let\labelitemiv\Frlabelitemiv
1638 \ifdim\labelwidthFB<\z@
1639 \settowidth{\labelwidthFB}{\FrenchLabelItem}%
1640 \fi
1641 \ifdim\listindentFB<\z@
1642 \ifdim\parindent=\z@
1643 \setlength{\listindentFB}{1.5em}%
1644 \else
1645 \setlength{\listindentFB}{\parindent}%
1646 \fi
1647 \fi
1648 }

```

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

```

1649 \def\enumerateFB{%
1650 \ifnum \@enumdepth >\thr@@\@toodeep\else
1651 \advance\@enumdepth\@ne
1652 \edef\@enumctr{enum\romannumeral\the\@enumdepth}%
1653 \expandafter
1654 \list
1655 \csname label\@enumctr\endcsname
1656 {\FB@listHsettings
1657 \usecounter\@enumctr\def\makelabel##1{\hss\llap{##1}}}%
1658 \fi
1659 }
1660 \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).

```

1661 \def\descriptionFB{%
1662 \list{}{\FB@listHsettings
1663 \labelwidth\z@
1664 \itemindent-\leftmargin
1665 \ifnum\@listdepth=1
1666 \advance\itemindent by \listindentFB
1667 \fi
1668 \let\makelabel\descriptionlabel}%
1669 }
1670 \let\enddescriptionFB\endlistORI

```

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

```

\bbf@frenchlistlayout1671 \def\update@frenchlists{%
\bbf@nonfrenchlistlayout1672 \ifFBReduceListSpacing \let\list\listFB \fi
1673 \ifFBStandardItemizeEnv
1674 \else \let\itemize\itemizeFB \fi
1675 \ifFBStandardItemLabels

```

```

1676 \else \labelitemsFB \fi
1677 \ifFBStandardEnumerateEnv
1678 \else \let\enumerate\enumerateFB \let\description\descriptionFB \fi
1679 }

```

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. 52.

```

1680 \def\FB@ufl{\relax}
1681 \def\bbl@frenchlistlayout{%
1682   \ifFBGlobalLayoutFrench
1683   \else
1684     \babel@save\list          \babel@save\itemize
1685     \babel@save\enumerate    \babel@save\description
1686     \babel@save\labelitemi   \babel@save\labelitemii
1687     \babel@save\labelitemiii \babel@save\labelitemiv
1688   \fi
1689   \FB@ufl
1690 }
1691 \def\bbl@nonfrenchlistlayout{%
1692   \ifFBGlobalLayoutFrench
1693     \update@frenchlists
1694   \fi
1695 }
1696 \FB@addto{extras}{\bbl@frenchlistlayout}
1697 \FB@addto{noextras}{\bbl@nonfrenchlistlayout}

```

## 2.11 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.

```

1698 \def\bbl@frenchindent{%
1699   \ifFBGlobalLayoutFrench\else\babel@save\@afterindentfalse\fi
1700   \ifFBIndentFirst
1701     \let\@afterindentfalse\@afterindenttrue
1702     \@afterindenttrue
1703   \fi}
1704 \def\bbl@nonfrenchindent{%
1705   \ifFBGlobalLayoutFrench
1706     \ifFBIndentFirst
1707       \@afterindenttrue
1708     \fi
1709   \fi}
1710 \FB@addto{extras}{\bbl@frenchindent}
1711 \FB@addto{noextras}{\bbl@nonfrenchindent}

```

## 2.12 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 2.9). 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.

```

1712 \AtBeginDocument{\ifpackageloaded{bigfoot}%
1713                 {\PackageInfo{frenchb.ldb}%
1714                 {bigfoot package in use.\MessageBreak
1715                 frenchb will NOT customise footnotes;\MessageBreak
1716                 reported}}%
1717                 {\let\@footnotemarkORI\@footnotemark
1718                 \def\@footnotemarkFB{\leavevmode\unskip\unkern
1719                 \,\@footnotemarkORI}%
1720                 \ifFBAutoSpaceFootnotes
1721                 \let\@footnotemark\@footnotemarkFB
1722                 \fi}%
1723                 }

```

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).

```

1724 \newcommand*{\dotFFN}{.}
1725 \newcommand*{\kernFFN}{\kern .5em}
1726 \newdimen\parindentFFN
1727 \parindentFFN=10in
1728 \def\ftnISSymbol{\@fnsymbol\c@footnote}
1729 \long\def\@makefntextFB#1{\ifx\thefootnote\ftnISSymbol
1730                          \@makefntextORI{#1}%
1731                          \else
1732                          \parindent=\parindentFFN
1733                          \rule{z@}{footnotesep}
1734                          \setbox\@tempboxa\hbox{\@thefnmark}%
1735                          \ifdim\wd\@tempboxa>z@

```

```

1736             \llap{\@thefnmark}\dotFFN\kernFFN
1737             \fi #1
1738         \fi}%

```

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

```

1739 \AtBeginDocument{\@ifpackageloaded{bigfoot}{}%
1740     {\ifdim\parindentFFN<10in
1741     \else
1742         \parindentFFN=\parindent
1743         \ifdim\parindentFFN<1.5em \parindentFFN=1.5em \fi
1744     \fi
1745     \let\@makefnORI\@makefn
1746     \long\def\@makefn#1{%
1747         \ifFBFrenchFootnotes
1748             \@makefnFB{#1}%
1749         \else
1750             \@makefnORI{#1}%
1751         \fi}%
1752     }%
1753 }

```

For compatibility reasons, we provide definitions for the commands dealing with the layout of footnotes in frenchb version 1.6. `\frenchbsetup{}` (see in section 2.9) 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 `\@makefn`.

```

1754 \newcommand*{\AddThinSpaceBeforeFootnotes}{\FBAutoSpaceFootnotestrue}
1755 \newcommand*{\FrenchFootnotes}{\FBFrenchFootnotestrue}
1756 \newcommand*{\StandardFootnotes}{\FBFrenchFootnotesfalse}

```

## 2.13 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.

```

1757 \FBclean@on@exit
1758 \ldf@finish\CurrentOption

```

### 3 Change History

v2.0	General: <code>\parindentFFN</code> not changed if already defined (required by JA for <code>cah-gut.cls</code> ). . . . .	62	and <code>StandardLayout</code> → <code>StandardLists</code> . . . . .	45
	Added warning for OT1 encoding. . . . .	56		
	Footnotes are now printed by default ‘à la française’ for the whole document. . . . .	62		
	New command <code>\frenchbsetup</code> added for global customisation. . . . .	45		
	<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		
	<code>\datefrench</code> : 2 ‘relax’ added in <code>\today</code> ’s definition. . . . .	34		
	<code>\FBtextellipsis</code> : Added special case for LY1 encoding, see bug report from Bruno Voisin (2004/05/18). . . . .	44		
	<code>\nombre</code> : <code>\nombre</code> now requires <code>numprint.sty</code> . . . . .	39		
v2.0b	General: Footnotes: Just do nothing (except warning) when the <code>bigfoot</code> package is loaded. . . . .	62		
v2.0c	General: There is no need to define here <code>numprint</code> ’s command <code>\npstylefrench</code> , it will be re-defined ‘AtBeginDocument’ by <code>\FBprocess@options</code> . . . . .	40		
	<code>\frenchbsetup</code> : Option <code>ThinSpaceInFrenchNumbers</code> added. . . . .	45		
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. . . . .	45		
v2.0e	<code>\frenchbsetup</code> : New option: <code>StandardLists</code> . . . . .	45		
v2.0f	<code>\frenchbsetup</code> : <code>StandardLayout</code> option had no effect on lists. Test moved to <code>\FBprocess@options</code> . . . . .	45		
	Two typos corrected in option <code>StandardLists</code> : <code>[false]</code> → <code>[true]</code>			
	<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> . . . . .	45		
v2.1a	General: Command <code>\fup</code> added to produce better superscripts than <code>\textsuperscript</code> . . . . .	35		
	<code>\datefrench</code> : <code>\today</code> changed (correction in 2.0 was wrong: <code>\today</code> was printed without spaces in toc). . . . .	34		
	<code>\frenchbsetup</code> : New option: <code>FrenchSuperscripts</code> to define <code>\up</code> as <code>\fup</code> or as <code>\textsuperscript</code> . . . . .	45		
	New option: <code>LowercaseSuperscripts</code> . . . . .	45		
v2.1b	General: Disable some commands in bookmarks. . . . .	56		
	<code>\fup</code> : Command <code>\fup</code> changed to use real superscripts from <code>fourier v. 1.6</code> . . . . .	35		
v2.1c	General: Added commands <code>\Nos</code> and <code>\nos</code> . . . . .	37		
	<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		
	<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		
v2.1d	General: Argument of <code>\ProvidesLanguage</code> changed above from ‘french’ to			



‘frenchb’ (otherwise <code>\listfiles</code> prints no date/version information). The real name of current language (french) as to be corrected before calling <code>\LdfInit</code> . 13	(suggested by JA). . . . . 62
Avoid warning “\end occurred when <code>\ifx ... incomplete</code> ” with LaTeX-2.09. . . . . 13	
v2.2a	v2.3c
<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> . . . . . 45	General: Commands <code>\ttfamily</code> , <code>\rmfamily</code> and <code>\sffamily</code> have to be robust. Bug introduced in 2.3a, pointed out by Manuel Pégourié-Gonnard. . . . . 29
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. . . . . 45	v2.3d
When frenchb is babel’s last option, French becomes the document’s main language, so GlobalLayout-French applies. . . . . 45	<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. 61
<code>\fup</code> : <code>\newif</code> and <code>\newdimen</code> moved before <code>\ifLaTeXe</code> to avoid an error with plainTeX. . . . . 35	v2.3e
v2.3a	General: Execute <code>\AutoSpaceBeforeFDP</code> also in LaTeX to define <code>\FDP@colonspace</code> : needed for tex4ht, pointed out by MPG. . . . 28
General: <code>\NoAutoSpaceBeforeFDP</code> and <code>\AutoSpaceBeforeFDP</code> now set the flag <code>\ifFBAutoSpacePunctuation</code> accordingly (LaTeX only). . . . . 28	v2.4a
In LaTeX, frenchb no longer adds spaces before ‘high punctuation’ characters in computer code. Suggested by Yannis Haralambous. 29	General: <code>\PackageWarning</code> changed to <code>\FBWarning</code> (when bigfoot package in use). . . . . 62
<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. . . . . 45	<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). . . . . 42
<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	<code>\frenchbsetup</code> : New option SuppressWarning. . . . . 45
v2.3b	<code>\ifFBXeTeX</code> : Added a new ‘if’ <code>\FBunicode</code> and some <code>\lccode</code> definitions to <code>\extrasfrench</code> and <code>\noextrasfrench</code> . . . . . 15
General: New commands <code>\dotFFN</code> and <code>\kernFFN</code> for more flexibility	v2.4c
	General: In <code>\ttfamilyFB</code> , also cancel automatic spaces inside French guillemets coded as characters (see <code>\frenchbsetup</code> ). . . . . 49
	In <code>\ttfamilyFB</code> , also cancel automatic spaces inside French guillemets entered as characters (see <code>\frenchbsetup</code> ). . . . . 29
	v2.4d
	<code>\up</code> : Command <code>\up</code> defined with <code>\providecommand</code> instead of

	\newcommand as \up may be defined elsewhere (catalan.ldf). Bug pointed out by Felip Manyé i Ballester. ....	35		\ifFBXeTeX: Added two new ‘if’ \FBXeTeX and \FBLuaTeX as XeTeX and behave differently regarding the status of the French “apostrophe”. ....	15
v2.5a	General: \og and \fg do not print correctly in English when using XeTeX or LuaTeX, fixed by using \textquotedblleft and \textquotedblright defined above. ....	31	v2.5e	General: \pdfstringdefDisableCommands should redefine \FB@og and \FB@fg instead of \og and \fg so that it works also when quotes are entered as characters. Reported by Sébastien Gouezel. ....	56
	New command \NoAutoSpacing, suggested by MPG. ....	29	v2.5f	General: Changed definitions of \at, \circonflexe, \tild, \boi and \degre for Unicode based engines. ....	37
	Punctuation is no longer made active with XeTeX-based engines. ....	16		\FBtextellipsis: Unicode fonts also provide a ready made character for \textellipsis, let’s just use it (reported by Maxime Chupin, 2011/06/04). ....	44
	\captionsfrench: \emph deleted in \seename and \alsoname to match what is done for the other languages. Suggested by Marc Baudoin. ....	40	v2.5g	General: Add four \newif to control spacing of quotes (characters and control sequences). ....	24
	\FBthinspace: Define \FBthinspace for those who want to customise the width of the space before ; and co. ....	17		Redefine \degre, \degres \at \circonflexe and \tild for bookmarks. Add \fup also. ....	56
	\textquoteddblright: Change \guillemotleft and \guillemotright definitions for Unicode and provide definitions for \textquotedblleft and \textquotedblright. Insures correct printing of quotes by \og and \fg in French and outside. ....	30		When \ifFB@xetex@punct is true, ‘og’ and ‘fg’ options now set XeTeXcharclasses of these characters to \FB@guilo and \FB@guilf. Otherwise French quotes behave as normal characters (their XeTeX-charclass is 0). ....	49
v2.5b	General: Do not use the test \iflanguage{french} 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 \l@french and \l@english are 0. Pointed out by Günter Milde. ....	46		\FB@xetex@punct@french: XeTeX-charclass(es) for French quotes will be set to \FB@guilo and \FB@guilf by options ‘og’ and ‘fg’ in \frenchbsetup. French quotes should behave as normal characters by default in XeLaTeX as in LaTeX. ....	25
v2.5c	General: The code meant for XeTeX also works for LuaTeX, we just need to change the test. ....	49	v2.5h	\degres: textcomp.sty has changed. The test about \M@TS1 is no longer relevant, let’s change it. ....	38
v2.5d	General: Moved the \newcount command outside \ifFB@xetex@punct ... \fi (it broke Plain formats). .	24	v2.5i	General: Temporary fix: as long as xeCJK.sty will not use	

\newXeTeXintercharclass to allocate its classes, we will have to define 3 fake classes. ....	24	\CaptionSeparator: Former \CaptionSeparator has been renamed as \FBCaption@Separator; Newif \ifFBwarning@capsep added. ....	42
\FB@xetex@punct@french: xeCJK.sty changes the \XeTeXcharclass 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 \FB@xetex@punct@nonfrench below. ....	25	v2.6d \FBthinspace: Rename \Fthinspace to \FBthinspace and \Fcolonspace to \FBcolonspace to avoid a conflict with fournier.sty. ....	17
v2.5j General: Previous fix removed: bug fixed in xeCJK.sty version 3.0.4 (06-May-2012). ....	24	v2.6e \degres: Refrain from redefining \textdegree from latin1.def, applemac.def, etc. as \degres because it loops in hyperref’s bookmarks. Pointed out by Eddy Flas on fctt. ....	38
v2.6a General: Bug correction: changing \leftmargin cannot be done only for itemize-lists: it messes up embedded enumerate lists. Pointed out by Denis Bitouzé. Lists have been completely redesigned in frenchb v. 2.6a. An option for backward compatibility is provided. ....	57	v2.6f \FB@itemizesettings: \labelwidth must be reset, f.i. when an itemize list occurs inside environments based on trivlist which set \labelwidth to 0 (see proof environment in amsthm.sty). Bug pointed out by Julien Hauseux. .	59
\frenchbsetup: New options ListOldLayout, StandardItemizeEnv and StandardEnumerateEnv (CompactItemize is deprecated). ....	45	v2.6g General: U+00A0 (Unicode nobreakspace) and U+202F (Unicode nobreakthinspace) added to class \FB@punctnul to prevent frenchb from adding it’s own space before ‘high punctuation’ characters. ....	24
\FrenchLabelItem: defaults to \textendash instead of \textendash up to v. 2.5k. ....	58	\FB@itemizesettings: Suppress all vertical spaces only if ReduceListSpacing is true. Pointed out by Pierre Willaime. ....	59
v2.6b \descriptionFB: Settings of \FB@listHsettings should apply to description lists too. ....	60	\ifFBXeTeX: lccode values for the French “apostrophe” are now the same for XeTeX and LuaTeX. ...	15
v2.6c General: Dummy file frenchb.cfg is no longer generated from frenchb.dtx. ....	12	v2.6h General: \FG@og and \FG@fg changed: former clumsy code removed. ...	49
No warning about \@makecaption for AMS classes. ....	42	If \@makecaption is undefined, no warning. ....	43
No warning about \@makecaption for koma-script classes. \captionformat customised in French. ....	43	New class \FB@guilnul for characters U+00A0 (Unicode nobreakspace) and U+202F (Unicode nobreakthinspace), to prevent frenchb from adding spurious spaces inside quotes. ....	24
Warning added when the caption or floatrow package is loaded before babel/frenchb. ....	43		

\CaptionSeparator: No active cat-	for captionnames. ....	40
codes in \STD@makecaption's def-	\datefrench: Take advantage of ba-	
inition. ....	bel's \SetString commands for	
42	\datefrench. Doesn't work with	
v3.0a	Plain (yet?). ....	34
General: \LdfInit checks	\descriptionFB: Add \listindentFB	
\datefrench instead of	to \itemindent. Suggested by De-	
\captionsfrench to avoid a con-	nis Bitouzé. ....	60
flict with papertex.cls which loads	\extrasfrench: Take advantage of	
datetime.sty. ....	babel's \babel@savevariable to	
13	handle apostrophe's \lccode. ..	15
\bbl@nonfrenchguillemets	\FBprocess@options: Changed op-	
deleted, use \babel@save in-	tion ThinColonSpace to make it	
stead. ....	work also with LuaTeX. ....	53
31	With koma-script and memoir class,	
Added explicit \FBguillskip for	customise \captionformat and	
LuaTeX. ....	\captiondelim. ....	54
31	\FBthinspace: LuaTeX requires	
Definitions of \FB@og and \FB@fg	dimensions: two new skips	
now depend on punctuation han-	\FBcolonskip and \FBthinspace. 17	
dling (LuaTeX / XeTeX / active). 31	\FBthinspace: \FBthinspace is no	
french.cfg will be loaded (if found)	longer a kern but a skip (frenchb	
instead of frenchb.cfg. NO NEED	adds a nobreak penalty before it). 17	
for .cfg files in French anyway. . 63	\frenchbsetup: New options Old-	
In Plain, provide a substi-	FigTabCaptions and Customise-	
tute for \PackageWarning and	FigTabCaptions. ....	45
\PackageInfo. ....	v3.0b	
14	General: frenchb.lua was not found	
Merging of \captionsfrenchb,	by Lua function dofile (not kpath-	
\captionsfrancais with	sea aware). Call function	
\captionsfrench deleted in fa-	kpse.find_file first, as suggested	
vor of new babel 3.9 syntax. ... 41	by Paul Gaborit. ....	23
More informative, less TeXnical	Require luatexbase with LaTeX in	
warning about \@makecaption. . 43	case fontspec has not been loaded	
New flag \ifFB@luatex@punct for	before babel. ....	17
'high punctuation' management	v3.0c	
with LuaTeX engines. ....	General: Activate option Standard-	
16	Lists when beamer class is	
New handling of 'high punctuation'	loaded. ....	46
through callbacks with LuaTeX en-	Bug correction: null glues should	
gines. ....	not trigger space insertion before	
17	high punctuation. Bug pointed out	
No warning about \@makecaption	by Benoit Rivet for the 'lstlisting'	
for SMF classes. No warning ei-	environment of the listings pack-	
ther with LuaTeX or XeTeX en-	age. ....	20
gines. ....	Changed \FBguill@spacing (inter-	
42	nal) to \FBguillspace (public). 31	
Options processing completely re-	frenchb requires babel-3.9i. ....	14
organised. ....	Just load luatexbase.sty instead of	
45	luaotfload.sty with plain formats. 17	
Support for options frenchb, fran-		
cais, canadien, acadian changed. 13		
Test \ifXeTeX changed to		
\ifFBunicode and 'xltextra'		
changed to 'fontspec'. ....		
56		
\CaptionSeparator: Remove		
\CaptionSeparatorORI, use		
\babel@save instead. ....		
42		
\captionsfrench: Take advantage		
of babel's \SetString commands		

Missing pair of braces after ‘and’ in the definition of ‘auto’. . . . .	20	package. . . . .	56
No need to define <code>\l@french</code> as <code>\lang@french</code> , <code>babel.def</code> (3.9j) takes care for this. . . . .	13	<code>frenchb.lua</code> : added flag <code>addgl</code> which must also be true when <code>prev</code> or <code>next</code> is not a char (i.e. <code>kern0</code> in <code>«\texttt{a}»</code> ). . . . .	21
<code>\datefrench</code> : <code>\SetString</code> still does not work for Plain with <code>babel 3.9k</code> . Need to define <code>\datefrench</code> . . .	34	<code>frenchb.lua</code> : codes <code>0x13</code> and <code>0x14</code> added for French quotes in T1-encoding. . . . .	18
<code>\frenchbsetup</code> : New option <code>InnerGuillSpace</code> . . . . .	45	<code>frenchb.lua</code> : look ahead when <code>next</code> is a kern (i.e in <code>« \texttt{a} »</code> ). . .	21
v3.1a		Misplaced <code>\fi</code> for plain formats. .	17
General: Codes "13 and "14 added for French quotes in T1-encoding. Support for older versions of LuaTeX and XeTeX dropped. . . . .	49	New command <code>\frquote</code> for imbedded or long French quotations. .	32
<code>fontspec</code> is not required for T1 fonts used be with the <code>luainputenc.sty</code>		<code>\frenchbsetup</code> : New options <code>InnerGuillSingle</code> , <code>EveryParGuill</code> and <code>EveryLineGuill</code> to control <code>\frquote</code> . .	45