

# A Babel language definition file for French

## frenchb.dtx v3.1e, 2015/04/10

Daniel Flipo  
daniel.flipo@free.fr

## Contents

<b>1 The French language</b>	<b>2</b>
1.1 Basic interface	2
1.2 Customisation	4
1.2.1 <code>\frenchbsetup</code>	4
1.2.2 Captions	8
1.3 Hyphenation checks	9
1.4 Changes	9
<b>2 The code</b>	<b>13</b>
2.1 Initial setup	13
2.2 Punctuation	16
2.2.1 Punctuation with LuaTeX	17
2.2.2 Punctuation with XeTeX	24
2.2.3 Punctuation with standard (pdf)TeX	27
2.2.4 Punctuation switches common to all engines	28
2.3 Commands for French quotation marks	30
2.4 Date in French	34
2.5 Extra utilities	35
2.6 Formatting numbers	39
2.7 Caption names	40
2.8 Dots...	44
2.9 More checks about packages' loading order	45
2.10 Setup options: keyval stuff	45
2.11 French lists	58
2.12 French indentation of sections	62
2.13 Formatting footnotes	63
2.14 Clean up and exit	64
<b>3 Change History</b>	<b>65</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 v3.1e are listed in subsection 1.4 p. 9.

An extensive documentation is available in French here:

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

## 1.1 Basic interface

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

`frenchb` takes account of babel’s *main language* defined as the *last* option at babel’s loading. When French is not babel’s main language, `frenchb` does not alter the general layout of the document (even in parts where French is the current language): the layout of lists, footnotes, indentation of first paragraphs of sections are not customised by `frenchb`.

When French is loaded as the last option of babel, `frenchb` makes the following changes to the global layout, *both in French and in all other languages*<sup>2</sup>:

1. the first paragraph of each section is indented (L<sup>A</sup>T<sub>E</sub>X only);
2. the default items in itemize environment are set to ‘—’ instead of ‘•’, and all vertical spacing and glue is deleted; it is possible to change ‘—’ to something else (‘-’ for instance) using `\frenchbsetup{}` (see section 1.2 p. 4);
3. vertical spacing in general L<sup>A</sup>T<sub>E</sub>X lists is shortened;
4. footnotes are displayed “à la française”.
5. the separator following the table or figure number in captions is printed as ‘-’ instead of ‘: ’; for changing this see 1.2.2 p. 8.

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

<sup>1</sup>The file described in this section has version number v3.1e and was last revised on 2015/04/10.

<sup>2</sup> For each item, hooks are provided to reset standard L<sup>A</sup>T<sub>E</sub>X settings or to emulate the behavior of former versions of `frenchb` (see command `\frenchbsetup{}`, section 1.2 p. 4).

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

1. French hyphenation patterns are made active;
2. ‘high punctuation’ characters (: ; ! ?) automatically add correct spacing in French; this is achieved using callbacks in Lua(La)TeX or ‘XeTeXinterchar’ mechanism in Xe(La)TeX; with TeX’82 and pdf(La)TeX these four characters are made active in the whole document;
3. `\today` prints the date in French;
4. the caption names are translated into French (L<sup>A</sup>T<sub>E</sub>X only). For customisation of caption names see section 1.2.2 p. 8.
5. the space after `\dots` is removed in French.

Some commands are provided by frenchb to make typesetting easier:

1. French quotation marks can be entered using the commands `\og` and `\fg` which work in L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> and PlainT<sub>E</sub>X, their appearance depending on what is available to draw them; even if you use L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> and T1-encoding, you should refrain from entering them as `<<~French quotation~>>`: `\og` and `\fg` provide better horizontal spacing (controlled by `\FBguillspace`). If French quote characters are available on your keyboard, you can use them, to get proper spacing in L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> see option `og=«`, `fg=»` p. 8.

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

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

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

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

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

2. A command `\up` is provided to typeset superscripts like `M\up{me}` (abbreviation for “Madame”), `1\up{er}` (for “premier”). Other commands are also provided for ordinals: `\ier`, `\iere`, `\iers`, `\ieres`, `\ieme`, `\iemes` (`3\iemes` prints 3<sup>es</sup>). All these commands take advantage of real superscript letters when they are available in the current font.
3. Family names should be typeset in small capitals and never be hyphenated, the macro `\bsc` (boxed small caps) does this, e.g., `L.\bsc{Lamport}` will print the same as `L.\mbox{\textsc{Lamport}}`. Note that composed names (such as Dupont-Durant) may now be hyphenated on explicit hyphens, this differs from frenchb v. 1.x.
4. Commands `\primo`, `\secundo`, `\tertio` and `\quarto` print 1<sup>o</sup>, 2<sup>o</sup>, 3<sup>o</sup>, 4<sup>o</sup>. `\FrenchEnumerate{6}` prints 6<sup>o</sup>.
5. Abbreviations for “Numéro(s)” and “numéro(s)” (N<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 documents written before 2008 without changes: by default `\up` now relies on `\fup` designed to produce better looking superscripts.

`AutoSpacePunctuation=false (true)` ; in French, the user *should* input a space before the four characters ‘:;!?’ but as many people forget about it (even among native French writers!), the default behaviour of frenchb is to automatically typeset 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 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  $\LaTeX$  classes only.

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

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

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

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

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

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

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

`og=«, fg=»` ; when guillemets characters are available on the keyboard (through a compose key for instance), it is nice to use them instead of typing `\og` and `\fg`. This option tells frenchb which characters are opening and closing French guillemets (they depend on the input encoding), then you can type either « guillemets » or «guillemets» (with or without spaces) to get properly typeset French quotes. This option works with LuaLaTeX and XeLaTeX; with pdfLaTeX it requires `inputenc` to be loaded with a proper encoding: 8-bits encoding (`latin1`, `latin9`, `ansinew`, `applemac`,...) or multi-byte encoding (`utf8`, `utf8x`).

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

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

### 1.2.2 Captions

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

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

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

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



### 1.3 Hyphenation checks

Once you have built your format, a good precaution would be to perform some basic tests about hyphenation in French. For  $\text{\LaTeX} 2_{\epsilon}$  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 \text{'}ev\text{'}enement alg\text{'}ebre}
\showhyphens{signal container événement algèbre}
\end{document}
```

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

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

Frequent mismatches:

- you get sig-nal con-tainer, this probably means that the hyphenation patterns you are using are for US-English, not for French;
- you get no hyphen at all in évé-ne-ment, this probably means that you are using CM fonts and the macro `\accent` to produce accented characters. 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 unbreakable 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  $\text{\LaTeX2}_{\epsilon}$  in compatibility mode has been dropped. This version is meant for  $\text{\LaTeX2}_{\epsilon}$  and Plain based formats (like `bplain`).  $\text{\LaTeX2}_{\epsilon}$  formats based on `\text{\LaTeX}` 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`. XeTeX  
**\ifBFLuaTeX** and LuaTeX engines require some extra code to deal with the French "apostrophe".  
**\ifFBXeTeX** Let's define three new 'if': `\ifBFLuaTeX`, `\ifFBXeTeX` and `\ifFBunicode` which will be true for XeTeX and LuaTeX engines and false for 8-bits engines.

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

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

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

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

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

```

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

```

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

```

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

```

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

```

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

```

## 2.2 Punctuation

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

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

```

129 \newif\ifFBAutoSpaceGuill \FBAutoSpaceGuilltrue

```

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

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

```

143 \newcommand*{\FBcolonspace}{\space}
144 \newcommand*{\FBthinspace}{\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@addGUIspace  \FB@addGUIspace=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 (beware of null values for fid, see \nullfont in TikZ):

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

239 end

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

```
240 local addDPspace    = luatexbase.attributes['FB@addDPspace']
241 local addGUILspace = luatexbase.attributes['FB@addGUILspace']
242 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`).

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

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

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

For characters for which `FB_punct_thin` or `FB_punct_thick` is *true*, the amount of spacing to be typeset before them is controlled by `\FBthinskip` (`thinwd`, `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 if attribute `\FB@addDPspace` is set, unless one of these three condition is met: a) the previous character is part of type `FB_punct_null` (this avoids spurious spaces in strings like `( ! )` or `??`), b) a null glue

(actually glues  $\leq 1$  sp for tabulars) precedes the punctuation character, c) the punctuation character starts a paragraph.

```

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

```

289     elseif SIG and SIG > 0 then
290         local addgl = (prev_char and not FB_guil_null[prev_char]) or
291                     (not prev_char)
292         if glue or addgl then
293             if glue then
294                 head = remove_node(head,prev,true)
295             end
296             local fbglue = new_glue_scaled(fid,guilwd,guilst,guilsh)
297             insert_node_before(head, item, copy_node(nobreak))
298             insert_node_before(head, item, copy_node(fbglue))
299         end
300     end
301 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.

```

302     if lang == FR and FB_punct_right[char] and fid > 0

```

```

303                                     and SIG and SIG > 0 then
304         local next = item.next
305         local next_id, next_subtype, next_char, kern_wd, nextnext
306         if next then
307             next_id = next.id
308             next_subtype = next.subtype
309             if next_id == GLYPH then
310                 next_char = next.char
311
312             A kern0 might hide a glue, so look ahead if next is a kern (this occurs with
313             « \texttt{a} »):
314             elseif next_id == KERN then
315                 kern_wd = next.kern
316                 if kern_wd == 0 then
317                     nextnext = next.next
318                     if nextnext then
319                         next = nextnext
320                         next_id = nextnext.id
321                         next_subtype = nextnext.subtype
322                         next_char = nextnext.char
323                     end
324                 end
325             end
326             local glue = next_id == GLUE and next_subtype == 0
327             if glue then
328                 glue_spec = next.spec
329                 glue_wd = glue_spec.width
330             end
331             glue = glue and glue_wd > 0
332             local addgl = (next_char and not FB_guil_null[next_char]) or
333                         (not next_char)
334             if glue or addgl then
335                 if glue then
336                     head = remove_node(head, next, true)
337                 end
338                 local fid = item.font
339                 local fbglue = new_glue_scaled(fid, guilwd, guilst, guilsh)
340                 insert_node_after(head, item, copy_node(fbglue))
341                 insert_node_after(head, item, copy_node(nobreak))
342             end
343         end
344     end
345     return head
346 end
347 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.

```

346 \ifFB@luatex@punct

```

```

347 \newcommand*{\FB@luatex@punct@french}{%
348   \ifx\shorthandoffORI\@undefined
349     \let\shorthandonORI\shorthandon
350     \let\shorthandoffORI\shorthandoff
351   \fi
352   \def\shorthandoff##1{%
353     \ifx\PackageWarning\@undefined
354       \fb@warning{\noexpand\shorthandoff{;!?} is helpless with
355         LuaTeX,\, use \noexpand\NoAutoSpacing
356         *inside a group* instead.}%
357     \else
358       \PackageWarning{frenchb.ldf}{\protect\shorthandoff{;!?} is
359         helpless with LuaTeX,\MessageBreak use \protect\NoAutoSpacing
360         \space *inside a group* instead;\MessageBreak reported}%
361     \fi}%
362   \def\shorthandon##1{}}%
363 }
364 \newcommand*{\FB@luatex@punct@nonfrench}{%
365   \ifx\shorthandoffORI\@undefined
366   \else
367     \let\shorthandon\shorthandonORI
368     \let\shorthandoff\shorthandoffORI
369   \fi
370 }
371 \FB@addto{extras}{\FB@luatex@punct@french}
372 \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).

```

373 \def\activate@luatexpunct{%
374   \directlua{%
375     FR = \the\l@french
376     local path = kpse.find_file("frenchb.lua", "lua")
377     if path then
378       local f = dofile(path)
379       luatexbase.add_to_callback("pre_linebreak_filter",
380         f, "frenchb.french_punctuation",1)
381       luatexbase.add_to_callback("hpack_filter",
382         f, "frenchb.french_punctuation",1)
383     else
384       texio.write_nl('')
385       texio.write_nl('*****')
386       texio.write_nl('Error: frenchb.lua not found.')
387       texio.write_nl('*****')
388       texio.write_nl('')
389     end
390   }%
391 }

```

392 \fi

End of specific code for punctuation with LuaTeX engines.

### 2.2.2 Punctuation with XeTeX

If `\XeTeXinterchartokenstate` is available, we use the “inter char” mechanism to provide correct spacing in French before the four characters ; ! ? and :. The basis of the following code was borrowed from the `polyglossia` package, see `gloss-french.ldf`. We use the same mechanism for French quotes (« and »), when automatic spacing for quotes is required by options `og=` and `fg=` in `\frenchbsetup{}` (see section 2.10).

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

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

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

```
393 \newcount\FB@interchartokenstateORI
394 \ifFB@xetex@punct
395   \ifLaTeXe
396     \PackageInfo{frenchb.ldf}{No need for active punctuation characters%
397       \MessageBreak with this version of XeTeX!%
398       \MessageBreak reported}
399   \else
400     \fb@info{No need for active punctuation characters\\
401       with this version of XeTeX!}
402   \fi
```

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

```
403 \ifx\@for\undefined
404   \def\@nnil{\@nil}%
405   \def\@empty{}%
406   \def\@fornoop#1\@#2#3{%
407     \long\def\@for#1:=#2\do#3{%
408       \expandafter\def\expandafter\@fortmp\expandafter{#2}%
409       \ifx\@fortmp\@empty \else
410         \expandafter\@forloop#2,\@nil,\@nil\@#1{#3}\fi%
411       \long\def\@forloop#1,#2,#3\@#4#5{\def#4{#1}\ifx #4\@nnil \else
412         #5\def#4{#2}\ifx #4\@nnil \else#5\@iforloop #3\@#4{#5}\fi\fi}%
413       \long\def\@iforloop#1,#2\@#3#4{\def#3{#1}\ifx #3\@nnil
414         \expandafter\@fornoop \else
415         #4\relax\expandafter\@iforloop\fi#2\@#3{#4}}%
416       \def\@tfor#1:={\@tfor#1 }%
417       \long\def\@tfor#1#2\do#3{\def\@fortmp{#2}\ifx\@fortmp\space\else
418         \@tforloop#2\@nil\@nil\@#1{#3}\fi}%
419       \long\def\@tforloop#1#2\@#3#4{\def#3{#1}\ifx #3\@nnil
```



```

420         \expandafter\@for\noop \else
421         #4\relax\expandafter\@tfor\loop\fi#2\@@#3{#4}}%
422 \fi

```

Six new character classes are defined for frenchb.

```

423 \newXeTeXintercharclass\FB@punctthick
424 \newXeTeXintercharclass\FB@punctthin
425 \newXeTeXintercharclass\FB@punctnul
426 \newXeTeXintercharclass\FB@guilo
427 \newXeTeXintercharclass\FB@guilf
428 \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.

```

429 \def\FB@charclassesORI{}
430 \def\empty{}
431 \def\FB@parse#1,#2\endparse{\def\FB@class{#1}%
432     \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.

```

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

```

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

```

434 \ifx\FB@charclassesORI\empty
435     \FB@interchartokenstateORI=\XeTeXinterchartokenstate
436     \@for\FB@char:={ '\: , '\; , '\! , '\? , "AB, "BB, %
437         '\( , '\[ , '\{ , '\, , '\. , '\- , '\) , '\} , %
438         '\%, "22, "27, "60, "2019, "A0, "202F }\do
439         {\edef\FB@charclassesORI{\FB@charclassesORI%
440             \theXeTeXcharclass\FB@char, }}%
441     \let\shorthandonORI\shorthandon
442     \let\shorthandoffORI\shorthandoff
443 \fi

```

Set the classes and interactions between classes.

```

444 \XeTeXinterchartokenstate=1
445 \XeTeXcharclass '\: = \FB@punctthick
446 \XeTeXinterchartoks \z@ \FB@punctthick = {%
447     \ifhmode\FDP@colonspace\fi}%
448 \XeTeXinterchartoks \FB@guilf \FB@punctthick = {%
449     \FDP@colonspace}%
450 \XeTeXinterchartoks 255 \FB@punctthick = {%
451     \ifhmode\unskip\penalty\@M\FBcolonspace\fi}%
452 \@for\FB@char:={ '\; , '\! , '\? }\do
453     {\XeTeXcharclass\FB@char=\FB@punctthin}%
454 \XeTeXinterchartoks \z@ \FB@punctthin = {%
455     \ifhmode\FDP@thinspace\fi}%
456 \XeTeXinterchartoks \FB@guilf \FB@punctthin = {%
457     \FDP@thinspace}%
458 \XeTeXinterchartoks 255 \FB@punctthin = {%

```



```

498     \let\shorthandoff\shorthandoffORI
499     \fi
500   }
501   \FB@addto{extras}{\FB@xetex@punct@french}
502   \FB@addto{noextras}{\FB@xetex@punct@nonfrench}

```

End of specific code for punctuation with modern XeTeX engines.

```
503 \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.

```

504 \ifFB@active@punct
505   \initiate@active@char{:}%
506   \initiate@active@char{;}%
507   \initiate@active@char{!}%
508   \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.

```

509   \declare@shorthand{french}{;}{;%
510     \ifhmode
511       \ifdim\lastskip>\z@
512         \unskip\penalty\M\FBthinspace
513       \else
514         \FDP@thinspace
515       \fi
516     \fi

```

Now we can insert a ; character.

```
517     \string;}

```

The next three definitions are very similar.

```

518   \declare@shorthand{french}{!}{;%
519     \ifhmode
520       \ifdim\lastskip>\z@
521         \unskip\penalty\M\FBthinspace
522       \else
523         \FDP@thinspace
524       \fi
525     \fi
526     \string!}
527   \declare@shorthand{french}{?}{;%
528     \ifhmode
529       \ifdim\lastskip>\z@
530         \unskip\penalty\M\FBthinspace

```

```

531         \else
532         \FDP@thinspace
533         \fi
534     \fi
535     \string?}
536 \declare@shorthand{french}{:}{}%
537     \ifhmode
538     \ifdim\lastskip>\z@
539     \unskip\penalty\M\FBcolonspace
540     \else
541     \FDP@colonspace
542     \fi
543 \fi
544 \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.

```

545 \declare@shorthand{system}{:}{\string:}
546 \declare@shorthand{system}{!}{\string!}
547 \declare@shorthand{system}{?}{\string?}
548 \declare@shorthand{system}{;}{\string;}
549 %}

```

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

```

550 \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.

```

551     \bbl@activate{:}\bbl@activate{;}%
552     \bbl@activate{!}\bbl@activate{?}%
553 }
554 \FB@addto{noextras}{%
555     \bbl@deactivate{:}\bbl@deactivate{;}%
556     \bbl@deactivate{!}\bbl@deactivate{?}%
557 }
558 \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.

```

559 \newif\ifFBAutoSpacePunctuation \FBAutoSpacePunctuationtrue

```

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

to 0 (false). User commands `\AutoSpaceBeforeFDP` and `\NoAutoSpaceBeforeFDP` do the same and take care of the flag `\ifFBAutoSpacePunctuation` in  $\LaTeX$ . Set the default now for Plain (done later for  $\LaTeX$ ).

```

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

```

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

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

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

```

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

```

```

595         \ifFB@luatex@punct\FB@addGUILspace=1 \fi
596         \ifFBAutoSpacePunctuation
597             \autospace@beforeFDP
598         \else
599             \noautospace@beforeFDP
600         \fi
601         \sffamilyORI}%
602 \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.

```

603 \newcommand*{\NoAutoSpacing}{\FBAutoSpaceGuillfalse
604     \ifFB@active@punct\shorthandoff{;:!?}\fi
605     \ifFB@xetex@punct\XeTeXinterchartokenstate=0 \fi
606     \ifFB@luatex@punct\FB@addDPspace=0 \FB@addGUILspace=0 \fi
607 }

```

## 2.3 Commands for French quotation marks

**\og** The top macros for quotation marks will be called **\og** (“ouvrez guillemets”) and **\fg** (“fermez guillemets”). Another option for typesetting quotes in French is to use the command **\frquote** (see below). Dummy definition of **\og** and **\fg** just to ensure that this commands are not yet defined. The default definition of **\og** and **\fg** will be set later (for English) by **\bbl@nonfrenchguillemets**.

```

608 \newcommand*{\og}{\@empty}
609 \newcommand*{\fg}{\@empty}

```

**\guillemotleft** **\guillemotright** **\textquotedblleft** **\textquotedblright** 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.

```

610 \ifLaTeXe
611 \else
612     \ifFBunicode
613         \def\guillemotleft{{\char"00AB}}
614         \def\guillemotright{{\char"00BB}}
615         \def\textquotedblleft{{\char"201C}}
616         \def\textquotedblright{{\char"201D}}
617     \else
618         \def\guillemotleft{\leavevmode\raise0.25ex
619             \hbox{$\scriptscriptstyle\ll$}}
620         \def\guillemotright{\raise0.25ex

```

```

621             \hbox{$\scriptscriptstyle\gg$}}
622   \def\textquotedblleft{''}
623   \def\textquotedblright{''}
624   \fi
625   \let\xspace\relax
626 \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 `lrm10` font, its dimensions will be scaled by `frenchb.lua` for the current font.

```

627 \newskip\FBguillskip
628 \FBguillskip=2.664pt plus 0.500pt minus 0.888pt \relax
629 \newcommand*{\FBguillspace}{\penalty\@M\hskip.8\fontdimen2\font
630                               plus.3\fontdimen3\font
631                               minus.8\fontdimen4\font}

```

`\FBguillspace` is not used with LuaTeX.

```

632 \ifFB@luatex@punct
633   \DeclareRobustCommand*{\FB@og}{\leavevmode
634     \bgroup\FB@addGUllspace=1 \guillemotleft\egroup}
635   \DeclareRobustCommand*{\FB@fg}{\ifdim\lastskip>\z@\unskip\fi
636     \bgroup\FB@addGUllspace=1 \guillemotright\egroup\xspace}
637 \fi

```

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

```

638 \ifFB@xetex@punct
639   \DeclareRobustCommand*{\FB@og}{\leavevmode
640     \bgroup\FBAutoSpaceGuillfalse\guillemotleft\egroup
641     \FBguillspace}
642   \DeclareRobustCommand*{\FB@fg}{\ifdim\lastskip>\z@\unskip\fi
643     \FBguillspace
644     \bgroup\FBAutoSpaceGuillfalse\guillemotright\egroup\xspace}
645 \fi
646 \ifFB@active@punct
647   \DeclareRobustCommand*{\FB@og}{\leavevmode
648     \guillemotleft
649     \FBguillspace}
650   \DeclareRobustCommand*{\FB@fg}{\ifdim\lastskip>\z@\unskip\fi
651     \FBguillspace
652     \guillemotright\xspace}
653 \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.

```

654 \ifLaTeXe
655   \def\bbf@frenchguillemets{\renewcommand*{\og}{\FB@og}%
656                               \renewcommand*{\fg}{\FB@fg}}
657   \renewcommand*{\og}{\textquotedblleft}
658   \renewcommand*{\fg}{\ifdim\lastskip>\z@ \unskip\fi \textquotedblright}
659 \else
660   \def\bbf@frenchguillemets{\let\og\FB@og
661                               \let\fg\FB@fg}
662   \def\og{\textquotedblleft}
663   \def\fg{\ifdim\lastskip>\z@ \unskip\fi \textquotedblright}
664 \fi
665 \FB@addto{extras}{\babel@save\og \babel@save\fg \bbf@frenchguillemets}

```

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

```

666 \newcommand*{\ogi}{\FB@og}
667 \newcommand*{\fgi}{\FB@fg}
668 \newcommand*{\ogii}{\textquotedblleft}
669 \newcommand*{\fgii}{\textquotedblright}

```

and the needed technical stuff to handle options:

```

670 \newcount\FBguill@level
671 \newif\ifFBcloseguill \FBcloseguilltrue
672 \newif\ifFBInnerGuillSingle
673 \def\FBguillopen{\bgroup\NoAutoSpacing\guillemotleft\egroup}
674 \def\FBguillclose{\bgroup\NoAutoSpacing\guillemotright\egroup}
675 \let\FBguillnone\relax
676 \let\FBeveryparguill\FBguillopen
677 \ifFB@luatex@punct
678   \let\FBeverylineguill\FBguillopen
679 \else
680   \let\FBeverylineguill\FBguillnone
681 \fi

```

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

```

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

```

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

```

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

```



```

690 \bgroup
691 \ifnum\FBguill@level=2
692 \PackageWarning{frenchb.ldf}{%
693 \protect\frquote\space accepts no more than two levels
694 \MessageBreak of quotations. Reported}
695 \else
696 \advance\FBguill@level by \@ne
697 \fi
698 \ifnum\FBguill@level=1

```

Set \FBeverypar@quote for level 1 quotations:

```

699 \ifx\FBeveryparguill\relax
700 \let\FBeverypar@quote\relax
701 \else
702 \def\FBeverypar@quote{\FBeveryparguill
703 \kern.8\fontdimen2\font}%
704 \fi
705 \logi
706 \everypar \expandafter{\the\everypar \FBeverypar@quote}%
707 #1\fgi
708 \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.

```

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

```

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

```

719 \let\FBeverypar@quote\relax
720 \ifFBInnerGuillSingle
721 \def\ogii{\leavevmode
722 \guilsinglleft\FBguillspace}%
723 \def\fgii{\ifdim\lastskip>\z@\unskip\fi
724 \FBguillspace\guilsinglright}%
725 \ifx\FBeveryparguill\FBguillopen
726 \def\FBeverypar@quote{\guilsinglleft
727 \kern.8\fontdimen2\font}%
728 \fi
729 \ifx\FBeveryparguill\FBguillclose
730 \def\FBeverypar@quote{\guilsinglright
731 \kern.8\fontdimen2\font}%
732 \fi

```

```

733         \fi
734         \ogii #1\ifFBcloseguill \fgii \fi
735     \fi
736 \fi
737 \fi
738 \egroup
739 }

```

## 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.

```

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

```

## 2.5 Extra utilities

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

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

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

```
775 \newif\ifFB@poorman
776 \newdimen\FB@Mht
777 \ifLaTeXe
778   \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{}`.

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

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.

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

814 \providecommand*\up{\relax}
      Poor man's definition of \up for Plain.
815 \else
816   \providecommand*\up[1]{\leavevmode\raise1ex\hbox{\sevenrm #1}}
817 \fi

```

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

```

      \ier 818 \def\ieme{\up{e}\xspace}
      \iere 819 \def\iemes{\up{es}\xspace}
      \iemes 820 \def\ier{\up{er}\xspace}
      \iers 821 \def\iers{\up{ers}\xspace}
      \ieres

```

```

822 \def\iere{\up{re}\xspace}
823 \def\ieres{\up{res}\xspace}

```

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

```

\no 824 \newcommand*{\FrenchEnumerate}[1]{%
\Nos 825      #1\up{o}\kern+.3em}
\nos 826 \newcommand*{\FrenchPopularEnumerate}[1]{%
\primo 827      #1\up{o}\kern+.3em}
\fprimo) Typing \primo should result in ‘1°’,

```

```

828 \def\primo{\FrenchEnumerate1}
829 \def\secundo{\FrenchEnumerate2}
830 \def\tertio{\FrenchEnumerate3}
831 \def\quarto{\FrenchEnumerate4}

```

while typing \fprimo) gives ‘1°’.

```

832 \def\fprimo{\FrenchPopularEnumerate1}
833 \def\fsecundo{\FrenchPopularEnumerate2}
834 \def\ftertio{\FrenchPopularEnumerate3}
835 \def\fquarto{\FrenchPopularEnumerate4}

```

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

```

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

```

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

```

840 \DeclareRobustCommand*{\bsc}[1]{\leavevmode\begin{group}\kern0pt
841      \scshape #1\endgroup}
842 \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.

```

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

```

```

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

```

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

```

```

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

```

## 2.6 Formatting numbers

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

```

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

```

`\nombre` The command `\nombre` is now borrowed from `numprint.sty` for L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>. There is no point to maintain the former tricky code when a package is dedicated to do the same job and more. For Plain based formats, `\nombre` no longer formats numbers, it prints them as is and issues a warning about the change.

Fake command `\nombre` for Plain based formats, warning users of frenchb v. 1.x. of the change.

```

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

```

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

```

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

```

```

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

```

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

```

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

```

## 2.7 Caption names

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

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

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

```

943 \ifx\FBfigtabshape\undefined \let\FBfigtabshape\scshape \fi

```

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

```

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

```



```

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

“Première partie” instead of “Part I”.

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

```

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

```

1004 \newif\if@FBwarning@capsep
1005 \@FBwarning@capseptrue
1006 \newcommand{\FBWarning}[2]{\PackageWarning{#1}{#2}}
1007 \newcommand*{\CaptionSeparator}{\space\textendash\space}
1008 \def\FBCaption@Separator{:\space}
1009 \long\def\FB@makecaption#1#2{%
1010   \vskip\abovcaptionskip
1011   \sbox\@tempboxa{#1\FBCaption@Separator #2}%
1012   \ifdim \wd\@tempboxa >\hsize
1013     #1\FBCaption@Separator #2\par
1014   \else
1015     \global \@minipagefalse
1016     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
1017   \fi
1018   \vskip\belowcaptionskip}

```

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

```

1019 \@ifclassloaded{amsart}{\@FBwarning@capsepfalse}{}
1020 \@ifclassloaded{amsbook}{\@FBwarning@capsepfalse}{}
1021 \@ifclassloaded{amsdtx}{\@FBwarning@capsepfalse}{}
1022 \@ifclassloaded{amslatex}{\@FBwarning@capsepfalse}{}
1023 \@ifclassloaded{amproc}{\@FBwarning@capsepfalse}{}
1024 \@ifclassloaded{smfart}{\@FBwarning@capsepfalse}{}
1025 \@ifclassloaded{smfbook}{\@FBwarning@capsepfalse}{}

```

Disable the standard warning unless high punctuation is active.

```

1026 \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.

```

1027 \newif\ifFB@koma
1028 \@ifclassloaded{memoir}{\@FBwarning@capsepfalse}{}
1029 \@ifclassloaded{scrartcl}{\@FBwarning@capsepfalse\FB@komatrue}{}
1030 \@ifclassloaded{scrbook}{\@FBwarning@capsepfalse\FB@komatrue}{}
1031 \@ifclassloaded{scrreprt}{\@FBwarning@capsepfalse\FB@komatrue}{}
1032 \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.

```

1033 \@ifpackageloaded{caption}
1034   {\FBWarning{frenchb.ldb}%
1035     {Please load the "caption" package\MessageBreak
1036       AFTER babel/frenchb; reported}%
1037   \@FBwarning@capsepfalse}%
1038   {}

```

Same for package floatrow.

```

1039 \@ifpackageloaded{floatrow}
1040   {\FBWarning{frenchb.ldb}%
1041     {Please load the "floatrow" package\MessageBreak
1042       AFTER babel/frenchb; reported}%
1043   \@FBwarning@capsepfalse}%
1044   {}

```

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').

```

1045 \AtBeginDocument{%
1046   \ifx\@makecaption\STD@makecaption
1047     \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.

```

1048   \ifFBoldFigTabCaptions
1049   \else
1050     \def\FBCaption@Separator{\iflanguage{french}{ : }{: }}%

```

```

1051 \fi
1052 \ifFBCustomiseFigTabCaptions
1053 \ifx\bbl@main@language\FB@french
1054 \def\FBCaption@Separator{\CaptionSeparator}%
1055 \fi
1056 \fi
1057 \@FBWarning@capsepfalse
1058 \fi
1059 \if@FBWarning@capsep
1060 \FBWarning{frenchb.ldb}%
1061 {Figures' and tables' captions might look like\MessageBreak
1062 'Figure 1:' which is wrong in French.\MessageBreak
1063 Check your class or packages to change this;\MessageBreak
1064 reported}%
1065 \fi
1066 \let\FB@makecaption\relax
1067 \let\STD@makecaption\relax
1068 }

```

## 2.8 Dots...

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

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

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

```

1069 \ifFBunicode
1070 \let\FBtextellipsis\textellipsis
1071 \else
1072 \DeclareTextSymbol{\FBtextellipsis}{LY1}{133}
1073 \DeclareTextCommandDefault{\FBtextellipsis}{%
1074 .\kern\fontdimen3\font.\kern\fontdimen3\font.\xspace}
1075 \fi

```

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

```

1076 \newcommand*{\Tdots@}{\@xp\textellipsis}
1077 \newcommand*{\Mdots@}{\@xp\mdots@}
1078 \AtBeginDocument{\DeclareRobustCommand*\dots}{\relax
1079 \csname\ifmmode M\else T\fi dots@\endcsname}%
1080 \ifdefined\@xp\else\let\@xp\relax\fi

```

```

1081             \ifdefined\mdots@ \else \let\Mdots@ \mathellipsis \fi
1082         }
1083 \def\bbl@frenchdots{\babel@save\Tdots@ \let\Tdots@\FBtextellipsis}
1084 \FB@addto{extras}{\bbl@frenchdots}

```

## 2.9 More checks about packages' loading order

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

```

1085 \ifFB@active@punct
1086   \ifpackageloaded{listings}
1087     {\FBWarning{frenchb.ldf}%
1088      {Please load the "listings" package\MessageBreak
1089       AFTER babel/frenchb; reported}%
1090     }{}
1091 \fi

```

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

```

1092 \newif\if@FBwarning@natbib
1093 \ifFB@active@punct
1094   \ifpackageloaded{natbib}{}{\@FBwarning@natbibtrue}
1095 \fi
1096 \AtBeginDocument{%
1097   \if@FBwarning@natbib
1098     \ifpackageloaded{natbib}{}{\@FBwarning@natbibfalse}%
1099   \fi
1100   \if@FBwarning@natbib
1101     \FBWarning{frenchb.ldf}%
1102     {Please load the "natbib" package\MessageBreak
1103      BEFORE babel/frenchb; reported}%
1104   \fi
1105 }

```

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

```

1106 \ifpackageloaded{numprint}
1107   {\ifnprt@autolanguage
1108     \FBWarning{frenchb.ldf}%
1109     {Please load the "numprint" package\MessageBreak
1110      AFTER babel/frenchb; reported}%
1111   \fi
1112 }{}

```

## 2.10 Setup options: keyval stuff

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

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

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

```
1113 \newcommand*{\frenchbsetup}[1]{%
1114   \setkeys{FB}{#1}%
1115 }%
1116 \@onlypreamble\frenchbsetup
```

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

```
1117 \newif\ifFBShowOptions           \FBShowOptionsfalse
1118 \newif\ifFBStandardLayout        \FBStandardLayouttrue
1119 \newif\ifFBGlobalLayoutFrench    \FBGlobalLayoutFrenchtrue
1120 \newif\ifFBReduceListSpacing     \FBReduceListSpacingfalse
1121 \newif\ifFBListOldLayout         \FBListOldLayoutfalse
1122 \newif\ifFBCompactItemize        \FBCompactItemizefalse
1123 \newif\ifFBStandardItemizeEnv     \FBStandardItemizeEnvtrue
1124 \newif\ifFBStandardEnumerateEnv  \FBStandardEnumerateEnvtrue
1125 \newif\ifFBStandardItemLabels    \FBStandardItemLabelstrue
1126 \newif\ifFBStandardLists        \FBStandardListstrue
1127 \newif\ifFBIndentFirst          \FBIndentFirstfalse
1128 \newif\ifFBFrenchFootnotes      \FBFrenchFootnotesfalse
1129 \newif\ifFBAutoSpaceFootnotes    \FBAutoSpaceFootnotesfalse
1130 \newif\ifFBOriginalTypewriter    \FBOriginalTypewriterfalse
1131 \newif\ifFBThinColonSpace        \FBThinColonSpacefalse
1132 \newif\ifFBThinSpaceInFrenchNumbers \FBThinSpaceInFrenchNumbersfalse
1133 \newif\ifFBFrenchSuperscripts    \FBFrenchSuperscriptstrue
1134 \newif\ifFBLowercaseSuperscripts \FBLowercaseSuperscriptstrue
1135 \newif\ifFBPartNameFull          \FBPartNameFulltrue
1136 \newif\ifFBCustomiseFigTabCaptions \FBCustomiseFigTabCaptionsfalse
1137 \newif\ifFBOldFigTabCaptions    \FBOldFigTabCaptionsfalse
1138 \newif\ifFBSmallCapsFigTabCaptions \FBSmallCapsFigTabCaptionstrue
1139 \newif\ifFBSuppressWarning       \FBSuppressWarningfalse
1140 \newif\ifFBINGuillSpace          \FBINGuillSpacefalse
```

The defaults values of these flags have been choosen 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.

```

1141 \edef\FB@french{\CurrentOption}
1142 \AtEndOfPackage{%
1143   \ifx\bbl@main@language\FB@french
1144     \FBGlobalLayoutFrenchtrue
1145     \@ifclassloaded{beamer}%
1146       {\PackageInfo{frenchb.ldf}{%
1147         No list customisation for the beamer class,%
1148         \MessageBreak reported}}%
1149       {\FBReduceListSpacingtrue
1150        \FBStandardItemizeEnvfalse
1151        \FBStandardEnumerateEnvfalse
1152        \FBStandardItemLabelsfalse}%
1153     \FBIndentFirsttrue
1154     \FBFrenchFootnotesttrue
1155     \FBAutoSpaceFootnotesttrue
1156     \FBCustomiseFigTabCaptionstrue
1157   \else
1158     \FBGlobalLayoutFrenchfalse
1159   \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.

```

1160 \RequirePackage{keyval}%
1161 \define@key{FB}{ShowOptions}[true]%
1162   {\csname FBShowOptions#1\endcsname}%
1163 \define@key{FB}{StandardLayout}[true]%
1164   {\csname FBStandardLayout#1\endcsname
1165    \ifFBStandardLayout
1166      \FBReduceListSpacingfalse
1167      \FBStandardItemizeEnvtrue
1168      \FBStandardItemLabelstrue
1169      \FBStandardEnumerateEnvtrue
1170      \FBIndentFirstfalse
1171      \FBFrenchFootnotesfalse
1172      \FBAutoSpaceFootnotesfalse
1173      \FBGlobalLayoutFrenchfalse
1174    \else
1175      \FBReduceListSpacingtrue
1176      \FBStandardItemizeEnvfalse
1177      \FBStandardItemLabelsfalse
1178      \FBStandardEnumerateEnvfalse
1179      \FBIndentFirsttrue
1180      \FBFrenchFootnotesttrue
1181      \FBAutoSpaceFootnotesttrue
1182    \fi}%
1183 \define@key{FB}{GlobalLayoutFrench}[true]%
1184   {\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.

```

1185             \ifFBGlobalLayoutFrench
1186             \ifx\bbbl@main@language\FB@french
1187             \else
1188             \PackageWarning{frenchb.ldb}%
1189             {Option 'GlobalLayoutFrench' skipped:%
1190             \MessageBreak French is *not*
1191             babel's last option.\MessageBreak}%
1192             \fi
1193             \fi}%
1194 \define@key{FB}{ReduceListSpacing}[true]%
1195     {\csname FBReduceListSpacing#1\endcsname}%
1196 \define@key{FB}{ListOldLayout}[true]%
1197     {\csname FBListOldLayout#1\endcsname
1198     \ifFBListOldLayout
1199     \FBStandardEnumerateEnvtrue
1200     \renewcommand*\FrenchLabelItem{\textendash}%
1201     \fi}%
1202 \define@key{FB}{CompactItemize}[true]%
1203     {\csname FBCompactItemize#1\endcsname
1204     \ifFBCompactItemize
1205     \FBStandardItemizeEnvfalse
1206     \FBStandardEnumerateEnvfalse
1207     \else
1208     \FBStandardItemizeEnvtrue
1209     \FBStandardEnumerateEnvtrue
1210     \fi}%
1211 \define@key{FB}{StandardItemizeEnv}[true]%
1212     {\csname FBStandardItemizeEnv#1\endcsname}%
1213 \define@key{FB}{StandardEnumerateEnv}[true]%
1214     {\csname FBStandardEnumerateEnv#1\endcsname}%
1215 \define@key{FB}{StandardItemLabels}[true]%
1216     {\csname FBStandardItemLabels#1\endcsname}%
1217 \define@key{FB}{ItemLabels}{%
1218     \renewcommand*\FrenchLabelItem{#1}}%
1219 \define@key{FB}{ItemLabeli}{%
1220     \renewcommand*\Frlabelitemi{#1}}%
1221 \define@key{FB}{ItemLabelii}{%
1222     \renewcommand*\Frlabelitemii{#1}}%
1223 \define@key{FB}{ItemLabeliii}{%
1224     \renewcommand*\Frlabelitemiii{#1}}%
1225 \define@key{FB}{ItemLabeliv}{%
1226     \renewcommand*\Frlabelitemiv{#1}}%
1227 \define@key{FB}{StandardLists}[true]%
1228     {\csname FBStandardLists#1\endcsname
1229     \ifFBStandardLists
1230     \FBReduceListSpacingfalse
1231     \FBCompactItemizefalse

```



```

1232             \FBStandardItemizeEnvtrue
1233             \FBStandardEnumerateEnvtrue
1234             \FBStandardItemLabelstrue
1235         \else
1236             \FBReduceListSpacingtrue
1237             \FBCompactItemizetrue
1238             \FBStandardItemizeEnvfalse
1239             \FBStandardEnumerateEnvfalse
1240             \FBStandardItemLabelsfalse
1241         \fi}%
1242 \define@key{FB}{IndentFirst}[true]%
1243         {\csname FBIndentFirst#1\endcsname}%
1244 \define@key{FB}{FrenchFootnotes}[true]%
1245         {\csname FBFrenchFootnotes#1\endcsname}%
1246 \define@key{FB}{AutoSpaceFootnotes}[true]%
1247         {\csname FBAutoSpaceFootnotes#1\endcsname}%
1248 \define@key{FB}{AutoSpacePunctuation}[true]%
1249         {\csname FBAutoSpacePunctuation#1\endcsname}%
1250 \define@key{FB}{OriginalTypewriter}[true]%
1251         {\csname FBOriginalTypewriter#1\endcsname}%
1252 \define@key{FB}{ThinColonSpace}[true]%
1253         {\csname FBThinColonSpace#1\endcsname}%
1254 \define@key{FB}{ThinSpaceInFrenchNumbers}[true]%
1255         {\csname FBThinSpaceInFrenchNumbers#1\endcsname}%
1256 \define@key{FB}{FrenchSuperscripts}[true]%
1257         {\csname FBFrenchSuperscripts#1\endcsname}%
1258 \define@key{FB}{LowercaseSuperscripts}[true]%
1259         {\csname FBLowercaseSuperscripts#1\endcsname}%
1260 \define@key{FB}{PartNameFull}[true]%
1261         {\csname FBPartNameFull#1\endcsname
1262         \ifFBPartNameFull
1263         \else
1264             \FB@addto{captions}{%
1265             \def\partname{Partie\protect\FB@Rpart}}}%
1266         \fi}%
1267 \define@key{FB}{CustomiseFigTabCaptions}[true]%
1268         {\csname FBCustomiseFigTabCaptions#1\endcsname}%
1269 \define@key{FB}{OldFigTabCaptions}[true]%
1270         {\csname FBOldFigTabCaptions#1\endcsname
1271
1272         \CurrentOption no longer defined. It's value has been saved in \FB@CurOpt while
1273         reading frenchb.ldf.
1274
1275         \ifFBOldFigTabCaptions
1276             \FB@addto{extras}{\babel@save\FBCaption@Separator
1277             \def\FBCaption@Separator{\CaptionSeparator}}}%
1278         \fi}%
1279 \define@key{FB}{SmallCapsFigTabCaptions}[true]%
1280         {\csname FBSmallCapsFigTabCaptions#1\endcsname
1281         \ifFBSmallCapsFigTabCaptions
1282             \let\FBfigtabshape\scshape
1283         \else

```

```

1280         \let\FBfigtabshape\relax
1281     \fi}%
1282 \define@key{FB}{SuppressWarning}[true]%
1283         {\csname FBSuppressWarning#1\endcsname
1284         \ifFBSuppressWarning
1285             \renewcommand{\FBWarning}[2]{\relax}%
1286         \fi}%

```

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

```

1287 \define@key{FB}{INGuillSpace}[true]%
1288         {\csname FBINGuillSpace#1\endcsname}%
1289 \define@key{FB}{InnerGuillSingle}[true]%
1290         {\csname FBInnerGuillSingle#1\endcsname}%
1291 \define@key{FB}{EveryParGuill}{\expandafter\let\expandafter
1292     \FBeveryparguill\csname FBguill#1\endcsname}%
1293 \define@key{FB}{EveryLineGuill}{\expandafter\let\expandafter
1294     \FBeverylineguill\csname FBguill#1\endcsname
1295     \ifFB@luatex@punct
1296     \else
1297         \let\FBeverylineguill\FBguillnone
1298         \PackageWarning{frenchb.ldf}%
1299             {Option 'EveryLineGuill' skipped:%
1300             \MessageBreak this option is for
1301             LuaTeX *only*.\MessageBreak Reported}%
1302 \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, apple mac, ...) 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.

```

1303 \define@key{FB}{og}{%
1304     \ifFBunicode

```

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

```

1305         \ifFB@luatex@punct
1306         \FB@addGUILLspace=1 \relax
1307     \fi

```

then with XeTeX it is a bit more tricky:

```

1308         \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).

```

1309         \XeTeXcharclass"13 = \FB@guilo
1310         \XeTeXcharclass"AB = \FB@guilo
1311         \XeTeXcharclass"A0 = \FB@guilnul
1312         \XeTeXcharclass"202F = \FB@guilnul
1313     \fi
1314 \else

```

This is for conventional TeX engines:

```

1315     \newcommand*\FB@@og{%
1316         \iflanguage{french}%
1317             {\ifFBAutoSpaceGuill\FB@og\ignorespaces
1318              \else\guillemotleft
1319              \fi}%
1320             {\guillemotleft}}%
1321     \AtBeginDocument{%
1322         \ifdefined\DeclareInputText
1323         \ifdefined\uc@dclc

```

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

```

1324         \uc@dclc{171}{default}{\FB@@og}%
1325     \else

```

if encoding is not utf8x, try utf8...

```

1326         \ifdefined\DeclareUnicodeCharacter
1327             utf8 loaded, use \DeclareUnicodeCharacter,
1328             \DeclareUnicodeCharacter{00AB}{\FB@@og}%
1329         \else
1330             \DeclareInputText{\the\@tempcnta}{\FB@@og}%
1331         \fi
1332     \fi
1333 \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.

```

1329         \@tempcnta'1\relax
1330         \ifdefined\mule@def
1331             \mule@def{11}{\FB@@og}%
1332         \else
1333             \DeclareInputText{\the\@tempcnta}{\FB@@og}%
1334         \fi
1335     \fi
1336 \fi
1337 \else

```

Package inputenc not loaded, no way...

```

1338     \PackageWarning{frenchb.ldf}%
1339         {Option 'og' requires package inputenc.\MessageBreak}%
1340     \fi
1341 }%
1342 \fi
1343 }%

```

Same code for the closing quote.

```

1344 \define@key{FB}{fg}{%
1345     \ifFBunicode
1346         \ifFB@luatex@punct
1347             \FB@addGUILspace=1 \relax
1348         \fi
1349         \ifFB@xetex@punct
1350             \XeTeXcharclass"14 = \FB@guilf
1351             \XeTeXcharclass"BB = \FB@guilf
1352             \XeTeXcharclass"A0 = \FB@guilnul
1353             \XeTeXcharclass"202F = \FB@guilnul
1354         \fi
1355     \else
1356         \newcommand*{\FB@@fg}{%
1357             \iflanguage{french}%
1358                 {\ifFBAutoSpaceGuill\FB@fg
1359                 \else\guillemotright
1360                 \fi}%
1361             {\guillemotright}}%
1362     \AtBeginDocument{%
1363         \ifdefined\DeclareInputText
1364             \ifdefined\uc@dcl
1365                 \uc@dcl{187}{default}{\FB@@fg}%
1366             \else
1367                 \ifdefined\DeclareUnicodeCharacter
1368                     \DeclareUnicodeCharacter{00BB}{\FB@@fg}%
1369                 \else
1370                     \@tempcnta'#1\relax
1371                     \ifdefined\mule@def
1372                         \mule@def{27}{\FB@@fg}%
1373                     \else
1374                         \DeclareInputText{\the\@tempcnta}{\FB@@fg}%
1375                     \fi
1376                 \fi
1377             \fi
1378         \else
1379             \PackageWarning{frenchb.ldf}%
1380                 {Option 'fg' requires package inputenc.\MessageBreak}%
1381         \fi
1382     }%
1383 \fi
1384 }%
1385 }

```

**\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.

```

1386 \newcommand*{\FBprocess@options}{%

```

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

```

1387 \@ifpackageloaded{enumitem}{%
1388   \ifFBStandardItemizeEnv
1389   \else
1390     \FBStandardItemizeEnvtrue
1391     \PackageInfo{frenchb.ldb}%
1392     {Setting StandardItemizeEnv=true for\MessageBreak
1393      compatibility with enumitem package,\MessageBreak}%
1394   \fi
1395   \ifFBStandardEnumerateEnv
1396   \else
1397     \FBStandardEnumerateEnvtrue
1398     \PackageInfo{frenchb.ldb}%
1399     {Setting StandardEnumerateEnv=true for\MessageBreak
1400      compatibility with enumitem package,\MessageBreak}%
1401   \fi}}%
1402 \@ifpackageloaded{paralist}{%
1403   \ifFBStandardItemizeEnv
1404   \else
1405     \FBStandardItemizeEnvtrue
1406     \PackageInfo{frenchb.ldb}%
1407     {Setting StandardItemizeEnv=true for\MessageBreak
1408      compatibility with paralist package,\MessageBreak}%
1409   \fi
1410   \ifFBStandardEnumerateEnv
1411   \else
1412     \FBStandardEnumerateEnvtrue
1413     \PackageInfo{frenchb.ldb}%
1414     {Setting StandardEnumerateEnv=true for\MessageBreak
1415      compatibility with paralist package,\MessageBreak}%
1416   \fi}}%
1417 \@ifpackageloaded{enumerate}{%
1418   \ifFBStandardEnumerateEnv
1419   \else
1420     \FBStandardEnumerateEnvtrue
1421     \PackageInfo{frenchb.ldb}%
1422     {Setting StandardEnumerateEnv=true for\MessageBreak
1423      compatibility with enumerate package,\MessageBreak}%
1424   \fi}}%

```

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

```

1425 \def\FB@ufl{\update@frenchlists}
1426 \ifx\bbl@main@language\FB@french
1427   \update@frenchlists
1428 \fi

```

The layout of footnotes is handled at the \begin{document} depending on the values of flags `FrenchFootnotes` and `AutoSpaceFootnotes` (see section 2.13), nothing has to be done here for footnotes.

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

```
1429 \ifBFAutoSpacePunctuation
1430     \autospace@beforeFDP
1431 \else
1432     \noautospace@beforeFDP
1433 \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.

```
1434 \ifFBOriginalTypewriter
1435 \else
1436     \let\ttfamilyORI\ttfamily
1437     \let\rmfamilyORI\rmfamily
1438     \let\sffamilyORI\sffamily
1439     \let\ttfamily\ttfamilyFB
1440     \let\rmfamily\rmfamilyFB
1441     \let\sffamily\sffamilyFB
1442 \fi
```

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

```
1443 \ifFBThinColonSpace
1444     \ifFB@luatex@punct
1445         \FBcolonskip=\FBthinskip\relax
1446     \else
1447         \renewcommand*{\FBcolonspace}{\FBthinspace}%
1448     \fi
1449 \fi
```

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

```
1450 \ifFBINGuillSpace
1451     \ifFB@luatex@punct
1452         \FBguillskip=3.33pt plus 1.665pt minus 1.11pt \relax
1453     \else
1454         \renewcommand*{\FBguillspace}{\space}%
1455     \fi
1456 \fi
```

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

```
1457 \@ifpackageloaded{numprint}%
1458 {\ifnprt@autolanguage
1459     \providecommand*{\npstylefrench}{}%
1460     \ifFBThinSpaceInFrenchNumbers
1461         \renewcommand*{\npstylefrench}%
1462             \npthousandsep{,}%
1463     }%
1464 }
```

```

1463         \npdecimalsign{,}%
1464         \npproductsign{\cdot}%
1465         \npunitseparator{\,}%
1466         \npdegreeseperator{ }%
1467         \nppercentseparator{\npert@unitsep}%
1468     }%
1469 \else
1470     \renewcommand*\npstylefrench{%
1471         \npthousandsep{~}%
1472         \npdecimalsign{,}%
1473         \npproductsign{\cdot}%
1474         \npunitseparator{\,}%
1475         \npdegreeseperator{ }%
1476         \nppercentseparator{\npert@unitsep}%
1477     }%
1478 \fi
1479 \npaddtolanguage{french}{french}%
1480 \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.

```

1481 \ifFBFrenchSuperscripts
1482     \DeclareRobustCommand*\up*{\@ifstar{\FB@up@fake}{\fup}}%
1483 \else
1484     \DeclareRobustCommand*\up*{\@ifstar{\FB@up@fake}%
1485                                     {\textsuperscript}}%
1486 \fi

```

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

```

1487 \ifFBLowercaseSuperscripts
1488 \else
1489     \renewcommand*\FB@lc[1]{##1}%
1490 \fi

```

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

```

1491 \ifFBCustomiseFigTabCaptions
1492     \ifFB@koma
1493         \let\captionformat\CaptionSeparator
1494     \fi
1495     \@ifclassloaded{memoir}%
1496         {\captiondelim{\CaptionSeparator}}{}%
1497 \else

```

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

```

1498 \ifFB@koma
1499     \def\captionformat{\iflanguage{french}{ : }{: }}%
1500 \fi
1501 \@ifclassloaded{memoir}%

```

```

1502         {\captiondelim{\iflanguage{french}{ : }{: }}{}}}%
1503     \fi

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

1504 \ifFBSShowOptions
1505     \GenericWarning{* }{%
1506         * **** List of possible options for frenchb ****\MessageBreak
1507         [Default values between brackets when frenchb is loaded *LAST*]%
1508         \MessageBreak
1509         ShowOptions=true [false]\MessageBreak
1510         StandardLayout=true [false]\MessageBreak
1511         GlobalLayoutFrench=false [true]\MessageBreak
1512         StandardLists=true [false]\MessageBreak
1513         IndentFirst=false [true]\MessageBreak
1514         ReduceListSpacing=false [true]\MessageBreak
1515         ListOldLayout=true [false]\MessageBreak
1516         StandardItemizeEnv=true [false]\MessageBreak
1517         StandardEnumerateEnv=true [false]\MessageBreak
1518         StandardItemLabels=true [false]\MessageBreak
1519         ItemLabels=\textendash, \textbullet,
1520             \protect\ding{43},... [\textendash]\MessageBreak
1521         ItemLabeli=\textendash, \textbullet,
1522             \protect\ding{43},... [\textendash]\MessageBreak
1523         ItemLabelii=\textendash, \textbullet,
1524             \protect\ding{43},... [\textendash]\MessageBreak
1525         ItemLabeliii=\textendash, \textbullet,
1526             \protect\ding{43},... [\textendash]\MessageBreak
1527         ItemLabeliv=\textendash, \textbullet,
1528             \protect\ding{43},... [\textendash]\MessageBreak
1529         FrenchFootnotes=false [true]\MessageBreak
1530         AutoSpaceFootnotes=false [true]\MessageBreak
1531         AutoSpacePunctuation=false [true]\MessageBreak
1532         OriginalTypewriter=true [false]\MessageBreak
1533         ThinColonSpace=true [false]\MessageBreak
1534         ThinSpaceInFrenchNumbers=true [false]\MessageBreak
1535         FrenchSuperscripts=false [true]\MessageBreak
1536         LowercaseSuperscripts=false [true]\MessageBreak
1537         PartNameFull=false [true]\MessageBreak
1538         SuppressWarning=true [false]\MessageBreak
1539         CustomiseFigTabCaptions=false [true]\MessageBreak
1540         OldFigTabCaptions=true [false]\MessageBreak
1541         SmallCapsFigTabCaptions=false [true]\MessageBreak
1542         INGuillSpace=true [false]\MessageBreak
1543         InnerGuillSingle=true [false]\MessageBreak
1544         EveryParGuill=open, close, none [open]\MessageBreak
1545         EveryLineGuill=open, close, none
1546             [open in LuaTeX, none otherwise]\MessageBreak
1547         og= <left quote character>, fg= <right quote character>%
1548         \MessageBreak
1549         *****%
1550         \MessageBreak\protect\frenchbsetup{ShowOptions}}

```



```

1551 \fi
1552 }

```

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.

```

1553 \AtBeginDocument{%
1554   \providecommand*\xspace{\relax}%

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

1555   \ifdefined\pdfstringdefDisableCommands
1556     \pdfstringdefDisableCommands{%
1557       \let\up\relax
1558       \let\up\relax
1559       \let\degre\textdegree
1560       \let\degres\textdegree
1561       \def\ieme{e\xspace}%
1562       \def\iemes{es\xspace}%
1563       \def\ier{er\xspace}%
1564       \def\iers{ers\xspace}%
1565       \def\iere{re\xspace}%
1566       \def\ieres{res\xspace}%
1567       \def\FrenchEnumerate#1{#1\degre\space}%
1568       \def\FrenchPopularEnumerate#1{#1\degre)\space}%
1569       \def\No{N\degre\space}%
1570       \def\no{n\degre\space}%
1571       \def\Nos{N\degre\space}%
1572       \def\nos{n\degre\space}%
1573       \def\FB@og{\guillemotleft\space}%
1574       \def\FB@fg{\space\guillemotright}%
1575       \def\at{@}%
1576       \def\circonflexe{\string^}%
1577       \def\tild{\string~}%
1578       \let\bsc\textsc
1579     }%
1580   \fi

```

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

```

1581 \FBprocess@options

```

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

```

1582 \ifFB@luatex@punct
1583   \activate@luatexpunct
1584 \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!

```

1585 \ifFBunicode
1586 \ifdefined\DeclareUTFcharacter
1587 \else
1588 \ifpackageloaded{luainputenc}{}%
1589 {\PackageWarning{frenchb.ldf}%
1590 {Add \protect\usepackage{fontspec} to the\MessageBreak
1591 preamble of your document,}%
1592 }%
1593 \fi
1594 \else
1595 \begingroup \newcommand{\FBOTone}{OT1}%
1596 \ifx\encodingdefault\FBOTone
1597 \PackageWarning{frenchb.ldf}%
1598 {OT1 encoding should not be used for French.%
1599 \MessageBreak
1600 Add \protect\usepackage[T1]{fontenc} to the
1601 preamble\MessageBreak of your document,}%
1602 \fi
1603 \endgroup
1604 \fi
1605 }

```

## 2.11 French lists

`\listFB` Vertical spacing in lists should be shorter in French texts than the defaults provided by  $\text{\LaTeX}$ . 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.

```

1606 \let\listORI\list
1607 \let\endlistORI\endlist
1608 \def\FB@listVsettings{%
1609 \setlength{\itemsep}{0.4ex plus 0.2ex minus 0.2ex}%
1610 \setlength{\parsep}{0.4ex plus 0.2ex minus 0.2ex}%
1611 \setlength{\topsep}{0.8ex plus 0.4ex minus 0.4ex}%
1612 \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`.

```

1613      \@tempdima=\parskip
1614      \addtolength{\topsep}{-\@tempdima}%
1615      \addtolength{\partopsep}{\@tempdima}%
1616 }
1617 \def\listFB#1#2{\listORI{#1}{\FB@listVsettings #2}}
1618 \let\endlistFB\endlist

```

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

- The ‘•’ is never used in French itemize-lists, an emdash ‘—’ or an en-dash ‘–’ is preferred for all levels. The item label to be used in French is stored in `\FrenchLabelItem`, it defaults to ‘—’ and can be changed using `\frenchbsetup{}` (see section 2.10).
- Vertical spacing between items, before and after the list, should be *null* with *no glue* added;
- In French the labels of itemize-lists are vertically aligned as follows:

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}
\FrenchLabelItem\newcommand*{\Frlabelitemi}{\FrenchLabelItem}
\FrenchLabelItem\newcommand*{\Frlabelitemii}{\FrenchLabelItem}
\FrenchLabelItem\newcommand*{\Frlabelitemiii}{\FrenchLabelItem}
\FrenchLabelItem\newcommand*{\Frlabelitemiv}{\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.

```

1624 \newlength\listindentFB
1625 \setlength{\listindentFB}{-1pt}
1626 \newlength\labelwidthFB
1627 \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.

```

1628 \newlength\leftmarginFB
1629 \def\FB@listHsettings{%
1630   \leftmarginFB\labelwidthFB
1631   \advance\leftmarginFB \labelsep

```

```

1632 \leftmargini\leftmarginFB
1633 \advance\leftmargini \listindentFB
1634 \leftmarginii\leftmarginFB
1635 \leftmarginiii\leftmarginFB
1636 \leftmarginiv\leftmarginFB
1637 \leftmargin\csname leftmargin\romannumeral\the\@listdepth\endcsname
1638 }

```

**\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).

```

1639 \def\FB@itemizesettings{%
1640   \ifFBReduceListSpacing
1641     \setlength{\itemsep}{\z@}%
1642     \setlength{\parsep}{\z@}%
1643     \setlength{\topsep}{\z@}%
1644     \setlength{\partopsep}{\z@}%
1645     \@tempdima=\parskip
1646     \addtolength{\topsep}{-\@tempdima}%
1647     \addtolength{\partopsep}{-\@tempdima}%
1648   \fi
1649   \settowidth{\labelwidth}{\csname\@itemitem\endcsname}%
1650   \ifFBListOldLayout
1651     \setlength{\leftmargin}{\labelwidth}%
1652     \addtolength{\leftmargin}{\labelsep}%
1653     \addtolength{\leftmargin}{\parindent}%
1654   \else
1655     \FB@listHsettings
1656   \fi
1657 }

```

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

```

1658 \def\itemizeFB{%
1659   \ifnum \@itemdepth >\thr@@\toodeep\else
1660     \advance\@itemdepth\@ne
1661     \edef\@itemitem{labelitem\romannumeral\the\@itemdepth}%
1662     \expandafter
1663     \listORI
1664     \csname\@itemitem\endcsname
1665     \FB@itemizesettings
1666   \fi
1667 }
1668 \let\enditemizeFB\endlistORI

1669 \def\labelitemsFB{%
1670   \let\labelitemi\Frlabelitemi
1671   \let\labelitemii\Frlabelitemii
1672   \let\labelitemiii\Frlabelitemiii

```

```

1673 \let\labelitemiv\Frlabelitemiv
1674 \ifdim\labelwidthFB<\z@
1675 \settowidth{\labelwidthFB}{\FrenchLabelItem}%
1676 \fi
1677 \ifdim\listindentFB<\z@
1678 \ifdim\parindent=\z@
1679 \setlength{\listindentFB}{1.5em}%
1680 \else
1681 \setlength{\listindentFB}{\parindent}%
1682 \fi
1683 \fi
1684 }

```

**\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`.

```

1685 \def\enumerateFB{%
1686 \ifnum \@enumdepth >\thr@@\@toodeep\else
1687 \advance\@enumdepth\@ne
1688 \edef\@enumctr{enum\romannumeral\the\@enumdepth}%
1689 \expandafter
1690 \list
1691 \csname label\@enumctr\endcsname
1692 {\FB@listHsettings
1693 \usecounter\@enumctr\def\makelabel##1{\hss\llap{##1}}}%
1694 \fi
1695 }
1696 \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).

```

1697 \def\descriptionFB{%
1698 \list{}{\FB@listHsettings
1699 \labelwidth\z@
1700 \itemindent-\leftmargin
1701 \ifnum\@listdepth=1
1702 \advance\itemindent by \listindentFB
1703 \fi
1704 \let\makelabel\descriptionlabel}%
1705 }
1706 \let\enddescriptionFB\endlistORI

```

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

```

\bbf@frenchlistlayout1707 \def\update@frenchlists{%
\bbf@nonfrenchlistlayout1708 \ifFBReduceListSpacing \let\list\listFB \fi
1709 \ifFBStandardItemizeEnv
1710 \else \let\itemize\itemizeFB \fi
1711 \ifFBStandardItemLabels

```

```

1712 \else \labelitemsFB \fi
1713 \ifFBStandardEnumerateEnv
1714 \else \let\enumerate\enumerateFB \let\description\descriptionFB \fi
1715 }

```

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. 53.

```

1716 \def\FB@ufl{\relax}
1717 \def\bbl@frenchlistlayout{%
1718   \ifFBGlobalLayoutFrench
1719   \else
1720     \babel@save\list          \babel@save\itemize
1721     \babel@save\enumerate     \babel@save\description
1722     \babel@save\labelitemi    \babel@save\labelitemii
1723     \babel@save\labelitemiii  \babel@save\labelitemiv
1724   \fi
1725   \FB@ufl
1726 }
1727 \def\bbl@nonfrenchlistlayout{%
1728   \ifFBGlobalLayoutFrench
1729     \update@frenchlists
1730   \fi
1731 }
1732 \FB@addto{extras}{\bbl@frenchlistlayout}
1733 \FB@addto{noextras}{\bbl@nonfrenchlistlayout}

```

## 2.12 French indentation of sections

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

```

1734 \def\bbl@frenchindent{%
1735   \ifFBGlobalLayoutFrench\else\babel@save\@afterindentfalse\fi
1736   \ifFBIndentFirst
1737     \let\@afterindentfalse\@afterindenttrue
1738     \@afterindenttrue
1739   \fi}
1740 \def\bbl@nonfrenchindent{%
1741   \ifFBGlobalLayoutFrench
1742     \ifFBIndentFirst
1743       \@afterindenttrue
1744     \fi
1745   \fi}
1746 \FB@addto{extras}{\bbl@frenchindent}
1747 \FB@addto{noextras}{\bbl@nonfrenchindent}

```

## 2.13 Formatting footnotes

The `bigfoot` package deeply changes the way footnotes are handled. When `bigfoot` is loaded, we just warn the user that `frenchb` will drop the customisation of footnotes. The layout of footnotes is controlled by two flags `\ifFBAutoSpaceFootnotes` and `\ifFBFrenchFootnotes` which are set by options of `\frenchbsetup{}` (see section 2.10). The layout of footnotes *does not depend* on the current language (just think of two footnotes on the same page looking different because one was called in a French part, the other one in English!).

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

```

1748 \AtBeginDocument{\@ifpackageloaded{bigfoot}%
1749                 {\PackageInfo{frenchb.ldb}%
1750                 {bigfoot package in use.\MessageBreak
1751                 frenchb will NOT customise footnotes;\MessageBreak
1752                 reported}}%
1753                 {\let\@footnotemarkORI\@footnotemark
1754                 \def\@footnotemarkFB{\leavevmode\unskip\unkern
1755                 \,\@footnotemarkORI}%
1756                 \ifFBAutoSpaceFootnotes
1757                 \let\@footnotemark\@footnotemarkFB
1758                 \fi}%
1759                 }

```

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).

```

1760 \newcommand*{\dotFFN}{.}
1761 \newcommand*{\kernFFN}{\kern .5em}
1762 \newdimen\parindentFFN
1763 \parindentFFN=10in
1764 \def\ftnISSymbol{\@fnsymbol\c@footnote}
1765 \long\def\@makefntextFB#1{\ifx\thefootnote\ftnISSymbol
1766                 \@makefntextORI{#1}%
1767                 \else
1768                 \parindent=\parindentFFN
1769                 \rule{z@}{\footnotesep}
1770                 \setbox\@tempboxa\hbox{\@thefnmark}%
1771                 \ifdim\wd\@tempboxa>z@

```

```

1772             \llap{\@thefnmark}\dotFFN\kernFFN
1773             \fi #1
1774             \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).

```

1775 \AtBeginDocument{\@ifpackageloaded{bigfoot}{}%
1776             {\ifdim\parindentFFN<10in
1777             \else
1778             \parindentFFN=\parindent
1779             \ifdim\parindentFFN<1.5em \parindentFFN=1.5em \fi
1780             \fi
1781             \let\@makefnORI\@makefn
1782             \long\def\@makefn#1{%
1783             \ifFBFrenchFootnotes
1784             \@makefnFB{#1}%
1785             \else
1786             \@makefnORI{#1}%
1787             \fi}%
1788             }%
1789             }

```

For compatibility reasons, we provide definitions for the commands dealing with the layout of footnotes in frenchb version 1.6. `\frenchbsetup{}` (see in section 2.10) should be preferred for setting these options. `\StandardFootnotes` may still be used locally (in minipages for instance), that's why the test `\ifFBFrenchFootnotes` is done inside `\@makefn`.

```

1790 \newcommand*{\AddThinSpaceBeforeFootnotes}{\FBAutoSpaceFootnotestrue}
1791 \newcommand*{\FrenchFootnotes}{\FBFrenchFootnotestrue}
1792 \newcommand*{\StandardFootnotes}{\FBFrenchFootnotesfalse}

```

## 2.14 Clean up and exit

Final cleaning. The macro `\ldf@finish` takes care for setting the main language to be switched on at `\begin{document}` and resetting the category code of `@` to its original value.

```

1793 \FBclean@on@exit
1794 \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> ). . . . . 63	v2.0g	<code>\frenchbsetup</code> : Revert previous change to <code>StandardLayout</code> . This option must set the three flags <code>\FBReduceListSpacingfalse</code> , <code>\FBCompactItemizefalse</code> , and <code>\FBStandardItemLabeltrue</code> instead of <code>\FBStandardListstrue</code> , so that later options can still change their value before executing <code>\FBprocess@options</code> . Same thing for option <code>StandardLists</code> . . . 46
	Added warning for OT1 encoding. . 57		
	Footnotes are now printed by default ‘à la française’ for the whole document. . . . . 63	v2.1a	General: Command <code>\fup</code> added to produce better superscripts than <code>\textsuperscript</code> . . . . . 35
	New command <code>\frenchbsetup</code> added for global customisation. . 45		<code>\datefrench</code> : <code>\today</code> changed (correction in 2.0 was wrong: <code>\today</code> was printed without spaces in toc). 34
	<code>\bsc</code> : <code>\hbox</code> dropped, replaced by <code>\kern0pt</code> . . . . . 37		<code>\frenchbsetup</code> : New option: French-Superscripts to define <code>\up</code> as <code>\fup</code> or as <code>\textsuperscript</code> . . . . . 46
	<code>\captionsfrench</code> : ‘Fig.’ changed to ‘Figure’ and ‘Tab.’ to ‘Table’. . . . 40		New option: LowercaseSuperscripts. . . . . 46
	<code>\datefrench</code> : 2 ‘ <code>\relax</code> ’ added in <code>\today</code> ’s definition. . . . . 34	v2.1b	General: Disable some commands in bookmarks. . . . . 57
	<code>\FBtextellipsis</code> : Added special case for LY1 encoding, see bug report from Bruno Voisin (2004/05/18). 44		<code>\fup</code> : Command <code>\fup</code> changed to use real superscripts from fourier v. 1.6. . . . . 35
	<code>\nombre</code> : <code>\nombre</code> now requires <code>numprint.sty</code> . . . . . 39	v2.1c	General: Added commands <code>\Nos</code> and <code>\nos</code> . . . . . 37
v2.0b	General: Footnotes: Just do nothing (except warning) when the bigfoot package is loaded. . . . . 63		<code>\degres</code> : Provide a temporary definition (hyperref safe) of <code>\degres</code> in case it has to be expanded in the preamble (by beamer’s <code>\title</code> command for instance). . . . . 38
v2.0c	General: There is no need to define here <code>numprint</code> ’s command <code>\npstylefrench</code> , it will be re-defined ‘ <code>AtBeginDocument</code> ’ by <code>\FBprocess@options</code> . . . . . 40		<code>\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
	<code>\frenchbsetup</code> : Option <code>ThinSpaceInFrenchNumbers</code> added. . . . . 46	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
v2.0d	<code>\frenchbsetup</code> : Options <code>og</code> and <code>fg</code> changed: limit the definition to French so that quote characters can be used in German. . . . . 46		
v2.0e	<code>\frenchbsetup</code> : New option: <code>StandardLists</code> . . . . . 46		
v2.0f	<code>\frenchbsetup</code> : <code>StandardLayout</code> option had no effect on lists. Test moved to <code>\FBprocess@options</code> . 46		
	Two typos corrected in option <code>StandardLists</code> : <code>[false] → [true]</code> and <code>StandardLayout → StandardLists</code> . 46		

(french) as to be corrected before calling <code>\LdfInit</code> . . . . .	13	<code>\rmfamily</code> and <code>\sffamily</code> have to be robust. Bug introduced in 2.3a, pointed out by Manuel Pégourié-Gonnard. . . . .	29
Avoid warning “\end occurred when <code>\ifx ... incomplete</code> ” with LaTeX-2.09. . . . .	13	v2.3d	
v2.2a		<code>\bbl@nonfrenchindent</code> : Bug correction: previous versions of <code>frenchb</code> 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. . . . .	62
<code>\frenchbsetup</code> : Default values of flags changed: default now means ‘StandardLayout’, they will be changed to ‘FrenchLayout’ <code>AtEndOfPackage</code> only if french is <code>\bbl@main@language</code> . . . . .	46	v2.3e	
The global layout of the document is no longer changed when <code>frenchb</code> is not the last option of <code>babel</code> ( <code>\bbl@main@language</code> ). Suggested by Ulrike Fischer. . . . .	46	General: Execute <code>\AutoSpaceBeforeFDP</code> also in LaTeX to define <code>\FDP@colonspace</code> : needed for <code>tex4ht</code> , pointed out by MPG. . . . .	28
When <code>frenchb</code> is <code>babel</code> ’s last option, French becomes the document’s main language, so <code>GlobalLayout-French</code> applies. . . . .	46	v2.4a	
<code>\fup</code> : <code>\newif</code> and <code>\newdimen</code> moved before <code>\ifLaTeXe</code> to avoid an error with <code>plainTeX</code> . . . . .	35	General: <code>\PackageWarning</code> changed to <code>\FBWarning</code> (when bigfoot package in use). . . . .	63
v2.3a		<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
General: <code>\NoAutoSpaceBeforeFDP</code> and <code>\AutoSpaceBeforeFDP</code> now set the flag <code>\ifFBAutoSpacePunctuation</code> accordingly (LaTeX only). . . . .	28	<code>\frenchbsetup</code> : New option <code>SuppressWarning</code> . . . . .	46
In LaTeX, <code>frenchb</code> no longer adds spaces before ‘high punctuation’ characters in computer code. Suggested by Yannis Haralambous. . . . .	29	<code>\ifFBXeTeX</code> : Added a new ‘if’ <code>\FBunicode</code> and some <code>\lccode</code> definitions to <code>\extrasfrench</code> and <code>\noextrasfrench</code> . . . . .	15
<code>\frenchbsetup</code> : New option: <code>OriginalTypewriter</code> . Now <code>frenchb</code> switches to <code>\noautospace@beforeFDP</code> when a <code>tt-font</code> is in use. When <code>OriginalTypewriter</code> is set to true, <code>frenchb</code> behaves as in pre-2.3 versions. . . . .	46	v2.4c	
<code>\fup</code> : <code>\lowercase</code> changed to <code>\MakeLowercase</code> as the former doesn’t work for non ASCII characters in encodings like <code>applemac</code> , <code>utf-8</code> ,... . . . .	35	General: In <code>\ttfamilyFB</code> , also cancel automatic spaces inside French guillemets coded as characters (see <code>\frenchbsetup</code> ). . . . .	50
v2.3b		In <code>\ttfamilyFB</code> , also cancel automatic spaces inside French guillemets entered as characters (see <code>\frenchbsetup</code> ). . . . .	29
General: New commands <code>\dotFFN</code> and <code>\kernFFN</code> for more flexibility (suggested by JA). . . . .	63	v2.4d	
v2.3c		<code>\up</code> : Command <code>\up</code> defined with <code>\providecommand</code> instead of <code>\newcommand</code> as <code>\up</code> may be defined elsewhere ( <code>catalan.ldf</code> ). Bug pointed out by Felip Manyé i Ballester. . . . .	35
General: Commands <code>\ttfamily</code> ,			



'{' '%' 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	to \FBthinspace and \Fcolonspace to \FBcolonspace to avoid a conflict with fournier.sty. ....	17
v2.5j		v2.6e	
General: Previous fix removed: bug fixed in xeCJK.sty version 3.0.4 (06-May-2012). ....	25	\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		v2.6f	
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. .	58	\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. ...	60
\frenchbsetup: New options ListOldLayout, StandardItemizeEnv and StandardEnumerateEnv (CompactItemize is deprecated). ....	46	v2.6g	
\FrenchLabelItem: default changed from \textendash to \textemdash. ....	59	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. ...	25
v2.6b		\FB@itemizesettings: Suppress all vertical spaces only if ReduceListSpacing is true. Pointed out by Pierre Willaime. ....	60
\descriptionFB: Settings of \FB@listHsettings should apply to description lists too. ....	61	\ifFBXeTeX: lccode values for the French "apostrophe" are now the same for XeTeX and LuaTeX. ....	15
v2.6c		v2.6h	
General: Dummy file frenchb.cfg is no longer generated from frenchb.dtx. ....	12	General: \FG@og and \FG@fg changed: former clumsy code removed. ...	50
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. ....	25
Warning added when the caption or floatrow package is loaded before babel/frenchb. ....	43	\CaptionSeparator: No active catcodes in \STD@makecaption's definition. ....	42
\CaptionSeparator: Former \CaptionSeparator has been renamed as \FBCaption@Separator; Newif \if@FBwarning@capsep added. ....	42	v3.0a	
v2.6d		General: \LdfInit checks \datefrench instead of \captionsfrench to avoid a conflict with papertex.cls which loads	
\FBthinspace: Rename \Fthinspace			

datetime.sty. ....	13	\extrasfrench: Take advantage of babel's \babel@savevariable to handle apostrophe's \lccode. ...	15
\bbl@nonfrenchguillemets deleted, use \babel@save in- stead. ....	32	\FBprocess@options: Changed op- tion ThinColonSpace to make it work also with LuaTeX. ....	54
Added explicit \FBguillskip for LuaTeX. ....	31	With koma-script and memoir class, customise \captionformat and \captiondelim. ....	55
Definitions of \FB@og and \FB@fg now depend on punctuation han- dling (LuaTeX / XeTeX / active). .	31	\FBthinspace: LuaTeX requires di- mensions: two new skips \FBcolonskip and \FBthinspace. 17	17
french.cfg will be loaded (if found) instead of frenchb.cfg. NO NEED for .cfg files in French anyway. ..	64	\FBthinspace: \FBthinspace is no longer a kern but a skip (frenchb adds a nobreak penalty before it). 17	17
In Plain, provide a substi- tute for \PackageWarning and \PackageInfo. ....	14	\frenchbsetup: New options Old- FigTabCaptions and Customise- FigTabCaptions. ....	46
Merging of \captionsfrenchb, \captionsfrancais with \captionsfrench deleted in favor of new babel 3.9 syntax. ....	41	v3.0b	
More informative, less TeXnical warning about \@makecaption. .	43	General: frenchb.lua was not found by Lua function dofile (not kpathsea aware). Call function kpse.find_file first, as suggested by Paul Ga- borit. ....	23
New flag \ifFB@luatex@punct for 'high punctuation' management with LuaTeX engines. ....	16	Require luatexbase with LaTeX in case fontspec has not been loaded before babel. ....	17
New handling of 'high punctuation' through callbacks with LuaTeX en- gines. ....	17	v3.0c	
No warning about \@makecaption for SMF classes. No warning either with LuaTeX or XeTeX engines. ..	42	General: Activate option StandardLists when beamer class is loaded. ...	47
Options processing completely reor- ganised. ....	45	Changed \FBguill@spacing (inter- nal) to \FBguillspace (public). .	31
Support for options frenchb, fran- cais, canadien, acadian changed. 13	13	frenchb requires babel-3.9i. ....	14
Test \ifXeTeX changed to \ifFBunicode and 'xltextra' changed to 'fontspec'. ....	57	frenchb.lua: null glues should not trigger space insertion before high punctuation. Bug pointed out by Benoit Rivet for the 'lstlisting' en- vironment of the listings package. 20	20
\CaptionSeparator: Remove \CaptionSeparatorORI, use \babel@save instead. ....	42	Just load luatexbase.sty instead of luaotfload.sty with plain formats. 17	17
\captionsfrench: Take advantage of babel's \SetString commands for captionnames. ....	40	No need to define \l@french as \lang@french, babel.def (3.9j) takes care for this. ....	13
\datefrench: Take advantage of ba- bel's \SetString commands for \datefrench. Doesn't work with Plain (yet?). ....	34	\datefrench: \SetString still does not work for Plain with babel 3.9k. Need to define \datefrench. ...	34
\descriptionFB: Add \listindentFB to \itemindent. Suggested by De- nis Bitouzé. ....	61	\frenchbsetup: New option IN- GuillSpace. ....	46
		v3.1a	
		General: Codes "13 and "14 added for French quotes in T1-encoding. Sup-	

port for older versions of LuaTeX and XeTeX dropped. ....	50	for \figurename and \tablename. ....	40
fontspec is not required for T1 fonts used with the luainputenc.sty package. ....	57	\fprimo): Removed \lowercase from definitions of \FrenchEnumerate, ... \No and co: \up already does the conversion. ....	37
frenchb.lua: added flag addgl which must also be true when prev or next is not a char (i.e. kern0 in «\texttt{a}»). ....	21	\frenchbsetup: New option SmallCapsFigTabCaptions. ....	46
frenchb.lua: codes 0x13 and 0x14 added for French quotes in T1-encoding. ....	18	\ieres: Removed \lowercase from definitions of \ieme and co: \up already does the conversion. ...	36
frenchb.lua: look ahead when next is a kern (i.e in « \texttt{a} »).	21	v3.1c	
Misplaced \fi for plain formats. ..	17	General: frenchb.lua: Previous bug fix for null glues (v3.0c) did not work properly. Fixed now (I hope). Pointed out by Jacques André. ..	20
New command \frquote for imbedded or long French quotations. ..	32	v3.1d	
\frenchbsetup: New options InnerGuillSingle, EveryParGuill and EveryLineGuill to control \frquote. ....	46	General: New section: issue warnings if packages listings, numprint and natbib are loaded too early or too late vs babel. ....	45
v3.1b		v3.1e	
General: frenchb.lua: add a check for null fid in french_punctuation (Tikz \nullfont). Bug pointed out by Paul Gaborit. ....	20	\frenchbsetup: Corrected typo: SmallCapsFigTabcaptions instead of SmallCapsFigTabCaptions. Pointed out by Céline Chevalier. .	46
\captionsfrench: Change \scshape to customisable \FBfigtabshape			