

A Babel language definition file for French

frenchb.dtx v3.1b, 2014/09/16

Daniel Flipo
daniel.flipo@free.fr

Contents

1 The French language	2
1.1 Basic interface	2
1.2 Customisation	4
1.2.1 <code>\frenchbsetup</code>	5
1.2.2 Captions	8
1.3 Hyphenation checks	9
1.4 Changes	9
2 The code	13
2.1 Initial setup	13
2.2 Punctuation	16
2.2.1 Punctuation with LuaTeX	17
2.2.2 Punctuation with XeTeX	23
2.2.3 Punctuation with standard (pdf)TeX	27
2.2.4 Punctuation switches common to all engines	28
2.3 Commands for French quotation marks	30
2.4 Date in French	34
2.5 Extra utilities	34
2.6 Formatting numbers	39
2.7 Caption names	40
2.8 Dots...	44
2.9 Setup options: keyval stuff	45
2.10 French lists	57
2.11 French indentation of sections	61
2.12 Formatting footnotes	62
2.13 Clean up and exit	63
3 Change History	64

1 The French language

The file `frenchb.dtx`¹, 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^AT_EX-2.09 is no longer supported. This new version (3.x) has been designed to be used only with L^AT_EX 2_ε and Plain formats based on TeX, pdfTeX, LuaTeX or XeTeX engines.

Changes between version 2.0 and v3.1b 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*²:

1. the first paragraph of each section is indented (L^AT_EX 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^AT_EX 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.

¹The file described in this section has version number v3.1b and was last revised on 2014/09/16.

² For each item, hooks are provided to reset standard L^AT_EX settings or to emulate the behavior of former versions of `frenchb` (see command `\frenchbsetup{}`, section 1.2 p. 4).

Regarding local typography, the command `\selectlanguage{french}` switches to the French language³, with the following effects:

1. French hyphenation patterns are made active;
2. ‘high punctuation’ characters (: ; ! ?) automatically add correct spacing in French; this is achieved using callbacks in Lua(La)TeX or ‘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^AT_EX 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^AT_EX_{2 ϵ} and PlainT_EX, their appearance depending on what is available to draw them; even if you use L^AT_EX_{2 ϵ} 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^AT_EX_{2 ϵ} see option `og=<<, fg=>>` p. 8.

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

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

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

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

³ `\selectlanguage{francais}` and `\selectlanguage{frenchb}` are no longer supported.

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

2. A command `\up` is provided to typeset superscripts like `M\up{me}` (abbreviation for “Madame”), `l\up{er}` (for “premier”). Other commands are also provided for ordinals: `\ier`, `\iere`, `\iers`, `\ieres`, `\ieme`, `\iemes` (`3\iemes` prints 3^{es}). 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{Lampport}` will print the same as `L.\mbox{\textsc{Lampport}}`. 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^o, 2^o, 3^o, 4^o. `\FrenchEnumerate{6}` prints 6^o.
5. Abbreviations for “Numéro(s)” and “numéro(s)” (N^o N^{os} n^o and n^{os}) are obtained via the commands `\No`, `\Nos`, `\no`, `\nos`.
6. Two commands are provided to typeset the symbol for “degré”: `\degre` prints the raw character and `\degres` should be used to typeset temperatures (e.g., “20~\degres C” with an 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_EXbook 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 `l\ier juin` will print ‘1^{er} juin’ (no need for a forced space after `l\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 ‘¹’, but has no effect on footnotes numbered with symbols (as in the `\thanks` command). Two commands `\StandardFootnotes` and `\FrenchFootnotes` are available to change the layout of footnotes locally; `\StandardFootnotes` can help when some footnotes are numbered with letters (inside minipages for instance).

`AutoSpaceFootnotes=false (true*)` ; by default frenchb adds a thin space in the running text before the number or symbol calling the footnote. Making this option `false` reverts to the standard setting (no space added).

`FrenchSuperscripts=false (true)` ; then `\up=\textsuperscript` (option added in version 2.1). Should only be made `false` to recompile older documents. By default `\up` now relies on `\fup` designed to produce better looking superscripts.

`AutoSpacePunctuation=false (true)` ; in French, the user *should* input a space before the four characters ‘; !?’ but as many people forget about it (even among native French writers!), the default behaviour of frenchb is to automatically typeset nobreakspaces the width of which is either `\FBthinspace` (defaults to `\thinspace`) before ‘;’ ‘!’ ‘?’ or `\FBcolonspace` (defaults to `\space`) before ‘:’; the defaults follow the French ‘Imprimerie Nationale’s recommendations. This is convenient in most cases but can lead to addition of spurious spaces in URLs, in MS-DOS paths or in timetables (10:55), except if they are typed in `\texttt` or verbatim mode. When the current font is a monospaced (typewriter) font, `AutoSpacePunctuation` is locally switched to `false`, no spurious space is added in that case, so the default behaviour of of frenchb in that area should be fine in most circumstances.

Choosing `AutoSpacePunctuation=false` will ensure that a proper space will be added before ‘:;!?’ *if and only if* a (normal) space has been typed in. Those who are unsure about their typing in this area should stick to the default option and use the provided `\NoAutoSpacing` command inside a group in case an unwanted space is added by frenchb (i.e. `{\NoAutoSpacing 10:55}`).

`ThinColonSpace=true (false)` changes the inter-word unbreakable space added before the colon ‘:’ to a thin space, so that the same amount of space is added before any of the four ‘high punctuation’ characters. The default setting is supported by the French ‘Imprimerie Nationale’.

`LowercaseSuperscripts=false (true)` ; by default frenchb inhibits the uppercasing of superscripts (for instance when they are moved to page headers). Making this option `false` will disable this behaviour (not recommended).

`PartNameFull=false (true)` ; when true, frenchb numbers the title of `\part{}` commands as “Première partie”, “Deuxième partie” and so on. With some classes which change the `\part{}` command (AMS classes do so), you could get “Première partie 1”, “Deuxième partie 2” in the toc; when this occurs, this option should be set to `false`, part titles will then be printed as “Partie I”, “Partie II”.

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

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

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

1.3 Hyphenation checks

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

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

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

- check the hyphenations proposed by T_EX 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.lfd`.

What's new in version 2.4?

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

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

Better compatibility with the `enumitem` package.

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

What's new in version 2.3?

Starting with version 2.3a, `frenchb` no longer inserts spaces automatically before ‘`;!?`’ when a typewriter font is in use; this was suggested by Yanniss 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 L^AT_EX command `\MakeLowercase` instead of T_EX’s `\lowercase` command. This prevents error messages when diacritics are used inside `\up{}` (diacritics should *never* be used in superscripts though!).

What's new in version 2.2?

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

What's new in version 2.1?

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

What's new in version 2.0?

Here is the list of all changes:

- Support for \LaTeX-2.09 and for $\text{\LaTeX}2_{\epsilon}$ in compatibility mode has been dropped. This version is meant for $\text{\LaTeX}2_{\epsilon}$ and Plain based formats (like `bplain`). $\text{\LaTeX}2_{\epsilon}$ formats based on m\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^AT_EX-2.09: issue a warning and exit if L^AT_EX-2.09 is in use. Plain is still supported.

```
35 \newif\ifLaTeXe
```

```

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

```

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

```

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

```

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

```

67 \let\bbbl@tempa\relax
68 \ifx\babeltags\undefined
69   \let\bbbl@tempa\endinput
70 \ifLaTeXe
71   \PackageError{frenchb.ldf}
72     {frenchb requires babel v.3.9i.\MessageBreak
73       Aborting here}
74     {Please upgrade Babel!}
75 \else
76   \fb@error{frenchb requires babel v.3.9i.\
77     Aborting here}
78     {Please upgrade Babel!}
79 \fi
80 \fi
81 \bbbl@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\bbbl@tempa{canadien}
93 \ifx\CurrentOption\bbbl@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`.
`\ifBLuaTeX` XeTeX and LuaTeX engines require some extra code to deal with the French “apostrophe”. Let’s define three new ‘if’: `\ifBLuaTeX`, `\ifBTeX` and `\ifFBUnicode` which will be true for XeTeX and LuaTeX engines and false for 8-bits engines. We cannot rely on ϵ -TeX’s `\ifdefined` at this stage, as it is not defined in Plain TeX format.

```

101 \newif\ifFBUnicode
102 \newif\ifBLuaTeX
103 \newif\ifBTeX
104 \begingroup\expandafter\expandafter\expandafter\endgroup
105 \expandafter\ifx\csname luatexversion\endcsname\relax
106 \else
107   \FBunicodetrue \BLuaTeXtrue
108 \fi
109 \begingroup\expandafter\expandafter\expandafter\endgroup
110 \expandafter\ifx\csname XeTeXrevision\endcsname\relax
111 \else
112   \Bunicodetrue \BTeXtrue
113 \fi

```

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

In French, character “apostrophe” is a letter in expressions like l’ambulance (French hyphenation patterns provide entries for this kind of words). This means that the `\lccode` of “apostrophe” has to be non null in French for proper hyphenation of those expressions, and has to be reset to null when exiting French. The following code ensures correct hyphenation of words like d’aventure, l’utopie, with all TeX engines (XeTeX, LuaTeX, pdfTeX) using `hyph-fr.tex` patterns.

```

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

```

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

```

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

```

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

```

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

```

2.2 Punctuation

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

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

```

129 \newif\ifFBAutoSpaceGuill \FBAutoSpaceGuilltrue

```

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

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

```

130 \newif\ifFB@active@punct \FB@active@puncttrue
131 \newif\ifFB@luatex@punct
132 \ifFBLuaTeX

```



```

133 \ifnum\luatexversion>75
134 \FB@luatex@puncttrue\FB@active@punctfalse
135 \fi
136 \fi

```

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

```

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

```

`\FBcolonspace` According to the I.N. specifications, the ‘:’ requires an inter-word space before it, `\FBthinspace` the other three require just a `\thinspace`. We define `\FBcolonspace` as `\space` (inter-word space) and `\FBthinspace` as `\thinspace` (both are user customisable). LuaTeX requires skips instead of commands, so we define `\FBcolonskip` and `\FBthinskip` to hold the specifications (width/stretch/shrink) of `\space` and `\thinspace` for the `lmr10` font; these parameters will be scaled for the current font by the `frenchb.lua` script (see how p. 19). `\FBcolonskip` and `\FBthinskip` are also user customisable.

```

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

```

2.2.1 Punctuation with LuaTeX

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

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

```

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

```

```

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

```

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

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

Local definitions for nodes:

```

196 local new_node = node.new

```

```

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

```

Some variables to store `\FBthinskip`, `\FBcolonskip` and `\FBguillskip` (given for `\lrm10`); `width/stretch/shrink` are stored as fractions of `\fontdimen2`, `\fontdimen3` and `\fontdimen4` of `\lrm10` 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 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       glue = glue and glue_wd > 0
```

For characters for which FB_punct_thin or FB_punct_thick is true, the amount of spacing to be typeset before them is controlled by \FBthinskip (thinwd, thinst, thinsh) or \FBcolonskip (colnwd, colnst, colnsh) respectively. Two options: if a space has been typed in before (turned to glue in the node list), we remove the glue and add a nobreak penalty and the required glue. Otherwise (auto option), the penalty and the required glue are inserted only if all three conditions are satisfied: a) attribute \FB@addDPspace is set, b) the previous character is not part of type FB_punct_null (this avoids spurious spaces in strings like (!) or ??), c) the punctuation mark does not start a paragraph.

```
266       if (FB_punct_thin[char] or FB_punct_thick[char]) then
267         local SBDP = has_attribute(item, addDPspace)
```

```

268     local fbglue
269     if FB_punct_thick[char] then
270         fbglue = new_glue_scaled(fid,colnwd,colnst,colnsh)
271     else
272         fbglue = new_glue_scaled(fid,thinwd,thinst,thinsh)
273     end
274     local auto =
275         SBDP and SBDP > 0 and
276         ((prev_char and not FB_punct_null[prev_char]) or
277         (not prev_char and (prev_id ~= 0 or prev_subtype ~= 3))
278         )
279     if glue or auto then
280         if glue then
281             head = remove_node(head,prev,true)
282         end
283         insert_node_before(head, item, copy_node(nobreak))
284         insert_node_before(head, item, copy_node(fbglue))
285     end

```

Let's consider '»' now (the only remaining glyph of FB_punct_left class): we just have to remove any *glue* possibly preceding '»', 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).

```

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

```

299     if lang == FR and FB_punct_right[char] and fid > 0
300         and SIG and SIG > 0 then
301         local next = item.next
302         local next_id, next_subtype, next_char, kern_wd, nextnext
303         if next then
304             next_id = next.id
305             next_subtype = next_subtype
306             if next_id == GLYPH then
307                 next_char = next.char

```

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

```

308     elseif next_id == KERN then
309         kern_wd = next.kern
310         if kern_wd == 0 then
311             nextnext = next.next
312             if nextnext then
313                 next = nextnext
314                 next_id = nextnext.id
315                 next_subtype = nextnext.subtype
316                 next_char = nextnext.char
317             end
318         end
319     end
320 end
321 local glue = next_id == GLUE and next_subtype == 0
322 if glue then
323     glue_spec = next.spec
324     glue_wd = glue_spec.width
325 end
326 glue = glue and glue_wd > 0
327 local addgl = (next_char and not FB_guil_null[next_char]) or
328              (not next_char)
329 if glue or addgl then
330     if glue then
331         head = remove_node(head,next,true)
332     end
333     local fid = item.font
334     local fbglue = new_glue_scaled(fid,guilwd,guilst,guilsh)
335     insert_node_after(head, item, copy_node(fbglue))
336     insert_node_after(head, item, copy_node(nobreak))
337 end
338 end
339 end
340 return head
341 end
342 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.

```

343 \ifFB@luatex@punct
344 \newcommand*{\FB@luatex@punct@french}{%
345 \ifx\shorthandoffORI\@undefined
346 \let\shorthandonORI\shorthandon
347 \let\shorthandoffORI\shorthandoff
348 \fi
349 \def\shorthandoff##1{%
350 \ifx\PackageWarning\@undefined
351 \fb@warning{\noexpand\shorthandoff{;:!?} is helpless with

```

```

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

```

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

```

End of specific code for punctuation with LuaTeX engines.

2.2.2 Punctuation with XeTeX

If `\XeTeXinterchartokenstate` is available, we use the “inter char” mechanism to provide correct spacing in French before the four characters ; ! ? and :. The

basis of the following code was borrowed from the polyglossia package, see `gloss-french.ldf`. We use the same mechanism for French quotes (« and »), when automatic spacing for quotes is required by options `og=` and `fg=` in `\frenchbsetup{}` (see section 2.9).

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

We switch `\XeTeXinterchartokenstate` to 1 and change the `\XeTeXcharclass` values of ; ! ? : (] « 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.

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

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

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

Six new character classes are defined for `frenchb`.

```
420 \newXeTeXintercharclass\FB@punctthick
421 \newXeTeXintercharclass\FB@punctthin
422 \newXeTeXintercharclass\FB@punctnul
423 \newXeTeXintercharclass\FB@guilo
```



```

424 \newXeTeXintercharclass\FB@guilf
425 \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.

```

426 \def\FB@charclassesORI{}
427 \def\empty{}
428 \def\FB@parse#1,#2\endparse{\def\FB@class{#1}%
429     \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.

```

430 \newcommand*\FB@xetex@punct@french{%
Saving must not be repeated if saved values are already in.
431 \if\FB@charclassesORI\empty
432 \FB@interchartokenstateORI=\XeTeXinterchartokenstate
433 \@for\FB@char:={'\:,\;\;,\!\;,\?,"AB,"BB,%
434     '\(,\[,'\{,\,,\.,'\-,\),'\},,%
435     '\%,"22,"27,"60,"2019,"A0,"202F}\do
436     {\edef\FB@charclassesORI{\FB@charclassesORI%
437         \theXeTeXcharclass\FB@char,}}%
438 \let\shorthandonORI\shorthandon
439 \let\shorthandoffORI\shorthandoff
440 \fi

```

Set the classes and interactions between classes.

```

441 \XeTeXinterchartokenstate=1
442 \XeTeXcharclass '\: = \FB@punctthick
443 \XeTeXinterchartoks \z@ \FB@punctthick = {%
444     \ifhmode\FDP@colonspace\fi}%
445 \XeTeXinterchartoks \FB@guilf \FB@punctthick = {%
446     \FDP@colonspace}%
447 \XeTeXinterchartoks 255 \FB@punctthick = {%
448     \ifhmode\unskip\penalty\@M\FBcolonspace\fi}%
449 \@for\FB@char:={'\;,\!\;,\?}\do
450     {\XeTeXcharclass\FB@char=\FB@punctthin}%
451 \XeTeXinterchartoks \z@ \FB@punctthin = {%
452     \ifhmode\FDP@thinspace\fi}%
453 \XeTeXinterchartoks \FB@guilf \FB@punctthin = {%
454     \FDP@thinspace}%
455 \XeTeXinterchartoks 255 \FB@punctthin = {%
456     \ifhmode\unskip\penalty\@M\FBthinspace\fi}%
457 \XeTeXinterchartoks \FB@guilo \z@ = {%
458     \ifFBAutoSpaceGuill\FBguillspace\fi}%
459 \XeTeXinterchartoks \FB@guilo 255 = {%
460     \ifFBAutoSpaceGuill\FBguillspace\ignorespaces\fi}%
461 \XeTeXinterchartoks \z@ \FB@guilf = {%
462     \ifFBAutoSpaceGuill\FBguillspace\fi}%
463 \XeTeXinterchartoks \FB@punctthin \FB@guilf = {%
464     \ifFBAutoSpaceGuill\FBguillspace\fi}%

```

```

465 \XeTeXinterchartoks 255 \FB@guilf = {%
466     \ifFBAutoSpaceGuill\unskip\FBguillspace\fi}%

```

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

```

467 \for\FB@char:={'\[, '\(, "A0, "202F}\do
468     {\XeTeXcharclass\FB@char=\FB@punctnul}%

```

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

```

469 \for\FB@char:={'\{, '\., '\., '\., '\., '\., '\., '\., '\., '\., '\., '\., '\., '\., '\., '\., \%
470     "22, "27, "60, "2019}\do
471     {\XeTeXcharclass\FB@char=\z@}%

```

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

```

472 \def\shorthandoff##1{%
473     \ifx\PackageWarning\@undefined
474         \fb@warning{\noexpand\shorthandoff{;:!?} is helpless with
475             XeTeX,\@ use \noexpand\NoAutoSpacing
476             *inside a group* instead.}%
477     \else
478         \PackageWarning{frenchb.lda}{\protect\shorthandoff{;:!?} is
479             helpless with XeTeX,\MessageBreak use \protect\NoAutoSpacing
480             \space *inside a group* instead;\MessageBreak reported}%
481     \fi}%
482 \def\shorthandon##1{%
483 }

```

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

```

484 \newcommand*\FB@xetex@punct@nonfrench{%
485     \ifx\FB@charclassesORI\empty
486     \else
487         \for\FB@char:={':, \;, \!, \?, "AB, "BB, %
488             '\(, '\[, '\{, '\., '\., '\., '\., '\., '\., '\., '\., \%
489             '\%, "22, "27, "60, "2019, "A0, "202F}\do
490             {\expandafter\FB@parse\FB@charclassesORI\endparse
491                 \XeTeXcharclass\FB@char=\FB@class}%
492         \def\FB@charclassesORI{%
493             \XeTeXinterchartokenstate=\FB@interchartokenstateORI
494             \let\shorthandon\shorthandonORI
495             \let\shorthandoff\shorthandoffORI
496         \fi
497     }
498     \FB@addto{extras}{\FB@xetex@punct@french}
499     \FB@addto{noextras}{\FB@xetex@punct@nonfrench}

```

End of specific code for punctuation with modern XeTeX engines.

```

500 \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.

```
501 \ifFB@active@punct
502 \initiate@active@char{:}%
503 \initiate@active@char{;}%
504 \initiate@active@char{!}%
505 \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.

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

Now we can insert a ; character.

```
514     \string;}
```

The next three definitions are very similar.

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

```

538         \FDP@colonspace
539         \fi
540     \fi
541     \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.

```

542 \declare@shorthand{system}{:}{\string;}
543 \declare@shorthand{system}{!}{\string!}
544 \declare@shorthand{system}{?}{\string?}
545 \declare@shorthand{system}{;}{\string;}
546 %}

```

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

```

547 \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.

```

548 \bbl@activate{:}\bbl@activate{;}%
549 \bbl@activate{!}\bbl@activate{?}%
550 }
551 \FB@addto{noextras}{%
552 \bbl@deactivate{:}\bbl@deactivate{;}%
553 \bbl@deactivate{!}\bbl@deactivate{?}%
554 }
555 \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.

```

556 \newif\ifFBAutoSpacePunctuation \FBAutoSpacePunctuationtrue

```

`\AutoSpaceBeforeFDP` `\autospace@beforeFDP` and `\noautospace@beforeFDP` are internal commands. `\NoAutoSpaceBeforeFDP` `\autospace@beforeFDP` defines `\FDP@thinspace` and `\FDP@colonspace` as unbreakable spaces and sets LuaTeX attribute `\FB@addDPspace` to 1 (true), while `\noautospace@beforeFDP` lets these spaces empty and sets flag `\FB@addDPspace` to 0 (false). User commands `\AutoSpaceBeforeFDP` and `\NoAutoSpaceBeforeFDP` do the same and take care of the flag `\ifFBAutoSpacePunctuation` in L^AT_EX. Set the default now for Plain (done later for L^AT_EX).

```

557 \def\autospace@beforeFDP{%
558     \ifFB@luatex@punct\FB@addDPspace=1 \fi
559     \def\FDP@thinspace{\penalty\@M\FBthinspace}%
560     \def\FDP@colonspace{\penalty\@M\FBcolonspace}}
561 \def\noautospace@beforeFDP{%
562     \ifFB@luatex@punct\FB@addDPspace=0 \fi

```

```

563         \let\FDP@thinspace\@empty
564         \let\FDP@colonspace\@empty}
565 \ifLaTeXe
566     \def\AutoSpaceBeforeFDP{\autospace@beforeFDP
567                             \FBAutoSpacePunctuationtrue}
568     \def\NoAutoSpaceBeforeFDP{\noautospace@beforeFDP
569                               \FBAutoSpacePunctuationfalse}
570     \AtEndOfPackage{\AutoSpaceBeforeFDP}
571 \else
572     \let\AutoSpaceBeforeFDP\autospace@beforeFDP
573     \let\NoAutoSpaceBeforeFDP\noautospace@beforeFDP
574     \AutoSpaceBeforeFDP
575 \fi

```

In L^AT_EX_{2 ϵ} `\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.

```

576 \ifLaTeXe
577     \DeclareRobustCommand\ttfamilyFB{%
578         \FBAutoSpaceGuillfalse
579         \ifFB@luatex@punct\FB@addGUILspace=0 \fi
580         \noautospace@beforeFDP\ttfamilyORI}%
581     \DeclareRobustCommand\rmfamilyFB{%
582         \FBAutoSpaceGuilltrue
583         \ifFB@luatex@punct\FB@addGUILspace=1 \fi
584         \ifFBAutoSpacePunctuation
585             \autospace@beforeFDP
586         \else
587             \noautospace@beforeFDP
588         \fi
589         \rmfamilyORI}%
590     \DeclareRobustCommand\sffamilyFB{%
591         \FBAutoSpaceGuilltrue
592         \ifFB@luatex@punct\FB@addGUILspace=1 \fi
593         \ifFBAutoSpacePunctuation
594             \autospace@beforeFDP
595         \else
596             \noautospace@beforeFDP
597         \fi
598         \sffamilyORI}%
599 \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.

```
600 \newcommand*{\NoAutoSpacing}{\FBAutoSpaceGuillfalse
601   \ifFB@active@punct\shorthandoff{;:!?}\fi
602   \ifFB@xetex@punct\XeTeXinterchartokenstate=0 \fi
603   \ifFB@luatex@punct\FB@addDPspace=0 \FB@addGUILspace=0 \fi
604 }
```

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`.

```
605 \newcommand*{\og}{\@empty}
606 \newcommand*{\fg}{\@empty}
```

\guillemotleft **\guillemotright** **\textquotedblleft** **\textquotedblright** L^AT_EX 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.

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

The next step is to provide correct spacing after `\guillemotleft` and before `\guillemotright`: a space precedes and follows quotation marks but no line break is allowed neither *after* the opening one, nor *before* the closing one. `\FBguillspace`

which does the spacing, has been fine tuned by Thierry Bouche to 80% of an inter-word space but with reduced stretchability. French quotes (including spacing) are printed by `\FB@og` and `\FB@fg`, the expansion of the top level commands `\og` and `\og` is different in and outside French. We'll try to be smart to users of David Carlisle's `xspace` package: if this package is loaded there will be no need for `{}` or `\` to get a space after `\fg`, otherwise `\xspace` will be defined as `\relax` (done at the end of this file).

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

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

`\FBguillspace` is not used with LuaTeX.

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

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

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

```
651 \ifLaTeXe
652   \def\bbL@frenchguillemets{\renewcommand*{\og}{\FB@og}%
653     \renewcommand*{\fg}{\FB@fg}}
654   \renewcommand*{\og}{\textquotedblleft}
655   \renewcommand*{\fg}{\ifdim\lastskip>\z@\unskip\fi \textquotedblright}
656 \else
```

```

657 \def\bbl@frenchguillemets{\let\og\FB@og
658 \let\fg\FB@fg}
659 \def\og{\textquotedblleft}
660 \def\fg{\ifdim\lastskip>\z@\unskip\fi\textquotedblright}
661 \fi
662 \FB@addto{extras}{\babel@save\og \babel@save\fg \bbl@frenchguillemets}

```

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

```

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

```

and the needed technical stuff to handle options:

```

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

```

The main command `\frquote` accepts (in $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2_{\epsilon}$ 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.

```

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

```

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

```

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

```


Set \FBeverypar@quote for level 1 quotations:

```

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

```

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

```

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

```

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

```

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

```

```

739   \fi
740   \egroup
741 }

```

2.4 Date in French

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

```

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

```

2.5 Extra utilities

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

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

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

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

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

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

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

The only packages I currently know to take advantage of real superscripts are a) `realscripts` used in conjunction with XeLaTeX or LuaLaTeX and OpenType fonts having the font feature ‘VerticalPosition=Superior’ and b) `fourier` (from version 1.6) when Expert Utopia fonts are available.

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

```
790 \def\FB@split#1#2#3#4\@nil{\def\FB@firstthree{#1#2#3}%
791                               \def\FB@suffix{#4}}
792 \def\FB@x{x}
```

```

793 \def\FB@j{j}
794 \DeclareRobustCommand*\FB@up}[1]{%
795   \bgroup \FB@poormantrue
796   \expandafter\FB@split\f@family\@nil

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

816 \providecommand*\up}{\relax}

```

Poor man's definition of \up for Plain.

```

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

```

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

```

\ier 820 \def\ieme{\up{e}\xspace}
\iere 821 \def\iemes{\up{es}\xspace}
\iemes 822 \def\ier{\up{er}\xspace}
\iers 823 \def\iers{\up{ers}\xspace}
\ieres 824 \def\iere{\up{re}\xspace}
      825 \def\ieres{\up{res}\xspace}

```

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

```
\no 826 \newcommand*{\FrenchEnumerate}[1]{%
\nos 827      #1\up{o}\kern+.3em}
\nos 828 \newcommand*{\FrenchPopularEnumerate}[1]{%
\primo 829      #1\up{o})\kern+.3em}
```

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

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

while typing `\fprimo` gives ‘1°’.

```
834 \def\fprimo{\FrenchPopularEnumerate1}
835 \def\fsecundo{\FrenchPopularEnumerate2}
836 \def\ftertio{\FrenchPopularEnumerate3}
837 \def\fquarto{\FrenchPopularEnumerate4}
```

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

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

```
842 \DeclareRobustCommand*\bsc[1]{\leavevmode\begin{group}\kern0pt
843      \scshape #1\end{group}}
844 \ifLaTeXe\else\let\scshape\relax\fi
```

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

```
845 \ifFBunicode
846   \newcommand*\at{{\char"0040}}
847   \newcommand*\circonflexe{{\char"005E}}
848   \newcommand*\tild{{\char"007E}}
849   \newcommand*\boi{\textbackslash}
850   \newcommand*\degre{{\char"00B0}}
851 \else
852   \ifLaTeXe
853     \DeclareTextSymbol{\at}{T1}{64}
854     \DeclareTextSymbol{\circonflexe}{T1}{94}
855     \DeclareTextSymbol{\tild}{T1}{126}
856     \DeclareTextSymbolDefault{\at}{T1}
857     \DeclareTextSymbolDefault{\circonflexe}{T1}
```

```

858 \DeclareTextSymbolDefault{\tild}{T1}
859 \DeclareRobustCommand*\boi{\textbackslash}
860 \DeclareRobustCommand*\degre{\r{}}
861 \else
862 \def\T@one{T1}
863 \ifx\fontencoding\T@one
864 \newcommand*\degre{{\char6}}
865 \else
866 \newcommand*\degre{{\char23}}
867 \fi
868 \newcommand*\at{{\char64}}
869 \newcommand*\circonflexe{{\char94}}
870 \newcommand*\tild{{\char126}}
871 \newcommand*\boi{{\backslash}}
872 \fi
873 \fi

```

`\degres` We now define a macro `\degres` 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 `\degres` to 0.3 em, this lets the symbol ‘degree’ stick to the preceding (e.g., 45\degres) or following character (e.g., 20~\degres C).

If T_EX Companion fonts are available (`textcomp.sty`), we pick up `\textdegree` from them instead of emulating ‘degrees’ from the `\r{}` accent. Otherwise we advise the user (once only) to use TS1-encoding.

```

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

```

2.6 Formatting numbers

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

```

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

```

`\nombre` The command `\nombre` is now borrowed from `numprint.sty` for L^AT_EX 2_ε. There is no point to maintain the former tricky code when a package is dedicated to do the same job and more. For Plain based formats, `\nombre` no longer formats numbers, it prints them as is and issues a warning about the change.

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

```

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

```

The next definitions only make sense for L^AT_EX 2_ε. For Plain based formats, let's activate LuaTeX punctuation if necessary, then cleanup and exit. Temporary fix: `\l@french` is not properly set by babel 3.9h with Plain LuaTeX format.

```

918 \let\FBstop@here\relax
919 \def\FBclean@on@exit{\let\ifLaTeXe\undefined
920     \let\LaTeXtrue\undefined
921     \let\LaTeXfalse\undefined}
922 \ifx\magnification\@undefined
923 \else
924   \def\FBstop@here{\ifFB@luatex@punct
925     \activate@luatexpunct
926     \fi
927   \FBclean@on@exit

```

```

928             \ldf@quit\CurrentOption\endinput}
929 \fi
930 \FBstop@here

```

What follows is for L^AT_εX 2_ε *only*; as all L^AT_εX 2_ε based formats include ϵ -T_εX, we can use `\ifdefined` now. We redefine `\nombre` for L^AT_εX 2_ε. A warning is issued at the first call of `\nombre` if `\numprint` is not defined, suggesting what to do. The package `numprint` is *not* loaded automatically by `frenchb` because of possible options conflict.

```

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

```

2.7 Caption names

The next step consists in defining the French equivalents for the L^AT_εX caption names.

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

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}`).

```

945 \ifx\FBfigtabshape\undefined \let\FBfigtabshape\scshape \fi

```

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

```

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

```



```

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

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

```

Up to v2.6h frenchb used to merge `\captionsfrenchb` and `\captionsfrançais` 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 LuaLaTeX and XeLaTeX, this glitch doesn’t occur, you get ‘Figure 1 :’ which is correct in French. With pdfLaTeX frenchb provides the following workaround. The standard definition of `\@makecaption` (e.g., the one provided in `article.cls`, `report.cls`, `book.cls` which is frozen for $\text{\LaTeX}2_{\epsilon}$ according to Frank Mittelbach), is saved in `\STD@makecaption`. ‘AtBeginDocument’ we compare it to its current definition (some classes like `memoir`, `koma-script` classes, `AMS` classes,

ua-thesis.cls...change it). If they are identical, frenchb just adds a hook called `\FBCaption@Separator` to `\@makecaption`; `\FBCaption@Separator` defaults to `' : '` as in the standard `\@makecaption` and will be changed to `' : '` in French `'AtBeginDocument'`; it can be also set to `\CaptionSeparator` (`' - '`) using [CustomiseFigTabCaptions](#).

While saving the standard definition of `\@makecaption` we have to make sure that characters `' :` and `' >` have `\catcode 12` (frenchb makes `' :` active and `spanish.lda` makes `' >` active).

```

993 \bgroup
994 \catcode':=12 \catcode'>=12 \relax
995 \long\gdef\STD@makecaption#1#2{%
996 \vskip\abovecaptionskip
997 \sbox\@tempboxa{#1: #2}%
998 \ifdim \wd\@tempboxa >\hsize
999 #1: #2\par
1000 \else
1001 \global \@minipagefalse
1002 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
1003 \fi
1004 \vskip\belowcaptionskip}
1005 \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.

```

1006 \newif\if@FBwarning@capsep
1007 \@FBwarning@capseptrue
1008 \newcommand{\FBWarning}[2]{\PackageWarning{#1}{#2}}
1009 \newcommand*\CaptionSeparator{\space\textendash\space}
1010 \def\FBCaption@Separator{: \space}
1011 \long\def\FB@makecaption#1#2{%
1012 \vskip\abovecaptionskip
1013 \sbox\@tempboxa{#1\FBCaption@Separator #2}%
1014 \ifdim \wd\@tempboxa >\hsize
1015 #1\FBCaption@Separator #2\par
1016 \else
1017 \global \@minipagefalse
1018 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
1019 \fi
1020 \vskip\belowcaptionskip}

```

Disable the standard warning with AMS and SMF classes.

```

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

```

```

1023 \@ifclassloaded{amsdtx}{\@FBwarning@capsepfalse}{}
1024 \@ifclassloaded{amslatex}{\@FBwarning@capsepfalse}{}
1025 \@ifclassloaded{amproc}{\@FBwarning@capsepfalse}{}
1026 \@ifclassloaded{smfart}{\@FBwarning@capsepfalse}{}
1027 \@ifclassloaded{smfbook}{\@FBwarning@capsepfalse}{}

```

Disable the standard warning unless high punctuation is active.

```

1028 \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.

```

1029 \newif\ifFB@koma
1030 \@ifclassloaded{memoir}{\@FBwarning@capsepfalse}{}
1031 \@ifclassloaded{scrartcl}{\@FBwarning@capsepfalse\FB@komatrue}{}
1032 \@ifclassloaded{scrbook}{\@FBwarning@capsepfalse\FB@komatrue}{}
1033 \@ifclassloaded{scrreprt}{\@FBwarning@capsepfalse\FB@komatrue}{}
1034 \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.

```

1035 \@ifpackageloaded{caption}
1036   {\FBWarning{frenchb.ldf}%
1037     {Please load the "caption" package\MessageBreak
1038       AFTER babel/frenchb; reported}%
1039   \@FBwarning@capsepfalse}%
1040   {}

```

Same for package floatrow.

```

1041 \@ifpackageloaded{floatrow}
1042   {\FBWarning{frenchb.ldf}%
1043     {Please load the "floatrow" package\MessageBreak
1044       AFTER babel/frenchb; reported}%
1045   \@FBwarning@capsepfalse}%
1046   {}

```

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').

```

1047 \AtBeginDocument{%
1048   \ifx\@makecaption\STD@makecaption
1049     \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.

```

1050   \ifFBoldFigTabCaptions
1051   \else
1052     \def\FBCaption@Separator{\iflanguage{french}{ : }{: }}%
1053   \fi
1054   \ifFBCustomiseFigTabCaptions

```

```

1055     \ifx\bbbl@main@language\FB@french
1056         \def\FBCaption@Separator{\CaptionSeparator}%
1057     \fi
1058 \fi
1059 \@FBwarning@capsepfalse
1060 \fi
1061 \if@FBwarning@capsep
1062     \FBWarning{frenchb.ldb}%
1063     {Figures' and tables' captions might look like\MessageBreak
1064     'Figure 1:' which is wrong in French.\MessageBreak
1065     Check your class or packages to change this;\MessageBreak
1066     reported}%
1067 \fi
1068 \let\FB@makecaption\relax
1069 \let\STD@makecaption\relax
1070 }

```

2.8 Dots...

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

The `\if` construction in the L^AT_EX 2_ε definition of `\dots` doesn't allow the use of `xspace` (`xspace` is always followed by a `\fi`), so we use the AMS-L^AT_EX 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.

```

1071 \ifFBunicode
1072     \let\FBtextellipsis\textellipsis
1073 \else
1074     \DeclareTextSymbol{\FBtextellipsis}{LY1}{133}
1075     \DeclareTextCommandDefault{\FBtextellipsis}{%
1076         .\kern\fontdimen3\font.\kern\fontdimen3\font.\xspace}
1077 \fi

```

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

```

1078 \newcommand*\Tdots@{\@xp\textellipsis}
1079 \newcommand*\Mdots@{\@xp\mdots@}
1080 \AtBeginDocument{\DeclareRobustCommand*\dots}{\relax
1081     \csname\ifmmode M\else T\fi dots@\endcsname}%
1082     \ifdefined\@xp\else\let\@xp\relax\fi
1083     \ifdefined\mdots@\else\let\Mdots@\mathellipsis\fi
1084 }

```

```

1085 \def\bb@frenchdots{\babel@save\Tdots@ \let\Tdots@\FBtextellipsis}
1086 \FB@addto{extras}{\bb@frenchdots}

```

2.9 Setup options: keyval stuff

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

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

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

```

1087 \newcommand*{\frenchbsetup}[1]{%
1088   \setkeys{FB}{#1}%
1089 }%
1090 \@onlypreamble\frenchbsetup

```

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

```

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

```

```

1113 \newif\ifFBSuppressWarning      \FBSuppressWarningfalse
1114 \newif\ifFBINGuillSpace        \FBINGuillSpacefalse

```

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

```

1115 \edef\FB@french{\CurrentOption}
1116 \AtEndOfPackage{%
1117   \ifx\bbl@main@language\FB@french
1118     \FBGlobalLayoutFrenchtrue
1119     \@ifclassloaded{beamer}%
1120     {\PackageInfo{frenchb.ldf}{%
1121       No list customisation for the beamer class,%
1122       \MessageBreak reported}}%
1123     {\FBReduceListSpacingtrue
1124      \FBStandardItemizeEnvfalse
1125      \FBStandardEnumerateEnvfalse
1126      \FBStandardItemLabelsfalse}%
1127     \FBIndentFirsttrue
1128     \FBFrenchFootnotesttrue
1129     \FBAutoSpaceFootnotesttrue
1130     \FBCustomiseFigTabCaptionstrue
1131   \else
1132     \FBGlobalLayoutFrenchfalse
1133   \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.

```

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

```

```

1151         \FBStandardItemLabelsfalse
1152         \FBStandardEnumerateEnvfalse
1153         \FBIndentFirsttrue
1154         \FBFrenchFootnotestru
1155         \FBAutoSpaceFootnotestru
1156         \fi}%
1157 \define@key{FB}{GlobalLayoutFrench}[true]%
1158         {\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.
1159         \ifFBGlobalLayoutFrench
1160         \ifx\bbbl@main@language\FB@french
1161         \else
1162         \PackageWarning{frenchb.ldb}%
1163         {Option 'GlobalLayoutFrench' skipped:%
1164         \MessageBreak French is *not*
1165         babel's last option.\MessageBreak}%
1166         \fi
1167         \fi}%
1168 \define@key{FB}{ReduceListSpacing}[true]%
1169         {\csname FBReduceListSpacing#1\endcsname}%
1170 \define@key{FB}{ListOldLayout}[true]%
1171         {\csname FBListOldLayout#1\endcsname
1172         \ifFBListOldLayout
1173         \FBStandardEnumerateEnvtrue
1174         \renewcommand*{\FrenchLabelItem}{\textendash}%
1175         \fi}%
1176 \define@key{FB}{CompactItemize}[true]%
1177         {\csname FBCompactItemize#1\endcsname
1178         \ifFBCompactItemize
1179         \FBStandardItemizeEnvfalse
1180         \FBStandardEnumerateEnvfalse
1181         \else
1182         \FBStandardItemizeEnvtrue
1183         \FBStandardEnumerateEnvtrue
1184         \fi}%
1185 \define@key{FB}{StandardItemizeEnv}[true]%
1186         {\csname FBStandardItemizeEnv#1\endcsname}%
1187 \define@key{FB}{StandardEnumerateEnv}[true]%
1188         {\csname FBStandardEnumerateEnv#1\endcsname}%
1189 \define@key{FB}{StandardItemLabels}[true]%
1190         {\csname FBStandardItemLabels#1\endcsname}%
1191 \define@key{FB}{ItemLabels}{%
1192         \renewcommand*{\FrenchLabelItem}{#1}}%
1193 \define@key{FB}{ItemLabeli}{%
1194         \renewcommand*{\Frlabelitemi}{#1}}%
1195 \define@key{FB}{ItemLabelii}{%
1196         \renewcommand*{\Frlabelitemii}{#1}}%
1197 \define@key{FB}{ItemLabeliii}{%

```

```

1198     \renewcommand*{\FrLabelitemiii}{#1}}%
1199 \define@key{FB}{ItemLabeliv}{%
1200     \renewcommand*{\FrLabelitemiv}{#1}}%
1201 \define@key{FB}{StandardLists}[true]%
1202     {\csname FBStandardLists#1\endcsname
1203     \ifFBStandardLists
1204         \FBReduceListSpacingfalse
1205         \FBCompactItemizefalse
1206         \FBStandardItemizeEnvtrue
1207         \FBStandardEnumerateEnvtrue
1208         \FBStandardItemLabelstrue
1209     \else
1210         \FBReduceListSpacingtrue
1211         \FBCompactItemizetrue
1212         \FBStandardItemizeEnvfalse
1213         \FBStandardEnumerateEnvfalse
1214         \FBStandardItemLabelsfalse
1215     \fi}%
1216 \define@key{FB}{IndentFirst}[true]%
1217     {\csname FBIndentFirst#1\endcsname}%
1218 \define@key{FB}{FrenchFootnotes}[true]%
1219     {\csname FBFrenchFootnotes#1\endcsname}%
1220 \define@key{FB}{AutoSpaceFootnotes}[true]%
1221     {\csname FBAutoSpaceFootnotes#1\endcsname}%
1222 \define@key{FB}{AutoSpacePunctuation}[true]%
1223     {\csname FBAutoSpacePunctuation#1\endcsname}%
1224 \define@key{FB}{OriginalTypewriter}[true]%
1225     {\csname FBOriginalTypewriter#1\endcsname}%
1226 \define@key{FB}{ThinColonSpace}[true]%
1227     {\csname FBThinColonSpace#1\endcsname}%
1228 \define@key{FB}{ThinSpaceInFrenchNumbers}[true]%
1229     {\csname FBThinSpaceInFrenchNumbers#1\endcsname}%
1230 \define@key{FB}{FrenchSuperscripts}[true]%
1231     {\csname FBFrenchSuperscripts#1\endcsname}
1232 \define@key{FB}{LowercaseSuperscripts}[true]%
1233     {\csname FBLowercaseSuperscripts#1\endcsname}
1234 \define@key{FB}{PartNameFull}[true]%
1235     {\csname FBPartNameFull#1\endcsname
1236     \ifFBPartNameFull
1237     \else
1238         \FB@addto{captions}{%
1239             \def\partname{Partie\protect\FB@Rpart}}%
1240     \fi}%
1241 \define@key{FB}{CustomiseFigTabCaptions}[true]%
1242     {\csname FBCustomiseFigTabCaptions#1\endcsname}%
1243 \define@key{FB}{OldFigTabCaptions}[true]%
1244     {\csname FBOldFigTabCaptions#1\endcsname
    \CurrentOption no longer defined. It's value has been saved in \FB@CurOpt while
    reading frenchb. ldf.
1245     \ifFBOldFigTabCaptions

```



```

1246         \FB@addto{extras}{\babel@save\FBCaption@Separator
1247             \def\FBCaption@Separator{\CaptionSeparator}}%
1248     \fi}%
1249 \define@key{FB}{SmallCapsFigTabCaptions}[true]%
1250     {\csname FBSmallCapsFigTabCaptions#1\endcsname
1251     \ifFBSmallCapsFigTabCaptions
1252         \let\FBfigtabshape\scshape
1253     \else
1254         \let\FBfigtabshape\relax
1255     \fi}%
1256 \define@key{FB}{SuppressWarning}[true]%
1257     {\csname FBSuppressWarning#1\endcsname
1258     \ifFBSuppressWarning
1259         \renewcommand{\FBWarning}[2]{\relax}%
1260     \fi}%

```

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

```

1261 \define@key{FB}{INGuillSpace}[true]%
1262     {\csname FBINGuillSpace#1\endcsname}%
1263 \define@key{FB}{InnerGuillSingle}[true]%
1264     {\csname FBInnerGuillSingle#1\endcsname}%
1265 \define@key{FB}{EveryParGuill}{\expandafter\let\expandafter
1266     \FBeveryparguill\csname FBguill#1\endcsname}%
1267 \define@key{FB}{EveryLineGuill}{\expandafter\let\expandafter
1268     \FBeverylineguill\csname FBguill#1\endcsname
1269     \ifFB@luatex@punct
1270     \else
1271         \let\FBeverylineguill\FBguillnone
1272         \PackageWarning{frenchb.lda}{%
1273             {Option 'EveryLineGuill' skipped:%
1274             \MessageBreak this option is for
1275             LuaTeX *only*.\MessageBreak Reported}}%
1276     \fi}%

```

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

With pdfTeX (or old LuaTeX and XeTeX engines), quote characters are made active and expand to `\og\ignorespaces` and `{\fg}` respectively if the current language is French, and to `\guillemotleft` and `\guillemotright` otherwise (think of German quotes), this is done by `\FB@@og` and `\FB@@fg`; thus correct unbreakable spaces will be added automatically to French quotes. The quote characters typed in depend on the input encoding, it can be single-byte (latin1, latin9, applemac, ...) or multi-bytes (utf-8, utf8x); the `inputenc` package has to be loaded before the `\begin{document}` with the proper coding option, so we check if `\DeclareInputText` is defined.

Life is much simpler here with modern LuaTeX or XeTeX engines: we just have to activate the `\FB@addGUIllSpace` attribute for LuaTeX or set `\XeTeXcharclass` of quotes to the proper value for XeTeX.

```

1277 \define@key{FB}{og}{%
1278     \ifFBunicode

```

```

LuaTeX or XeTeX in use, first try modern LuaTeX: we just need to set LuaTeX's
attribute \FB@addGUILspace to 1,
1279         \iffB@luatex@punct
1280         \FB@addGUILspace=1 \relax
1281         \fi
then with XeTeX it is a bit more tricky:
1282         \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 subsec-
tion 2.2).
1283         \XeTeXcharclass"13 = \FB@guilo
1284         \XeTeXcharclass"AB = \FB@guilo
1285         \XeTeXcharclass"A0 = \FB@guilnul
1286         \XeTeXcharclass"202F = \FB@guilnul
1287         \fi
1288         \else
This is for conventional TeX engines:
1289         \newcommand*\FB@@og{%
1290         \iflanguage{french}%
1291         {\iffBAutoSpaceGuill\FB@og\ignorespaces
1292         \else\guillemotleft
1293         \fi}%
1294         {\guillemotleft}}%
1295         \AtBeginDocument{%
1296         \ifdefined\DeclareInputText
1297         \ifdefined\uc@dclc
Package inputenc with utf8x encoding loaded, use \uc@dclc,
1298         \uc@dclc{171}{default}{\FB@@og}%
1299         \else
if encoding is not utf8x, try utf8...
1300         \ifdefined\DeclareUnicodeCharacter
utf8 loaded, use \DeclareUnicodeCharacter,
1301         \DeclareUnicodeCharacter{00AB}{\FB@@og}%
1302         \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.
1303         \@tempcnta'#1\relax
1304         \ifdefined\mule@def
1305         \mule@def{11}{\FB@@og}%
1306         \else
1307         \DeclareInputText{\the\@tempcnta}{\FB@@og}%
1308         \fi
1309         \fi
1310         \fi
1311         \else

```

Package inputenc not loaded, no way...

```
1312         \PackageWarning{frenchb.ldb}%  
1313         {Option 'og' requires package inputenc.\MessageBreak}%  
1314     \fi  
1315     }%  
1316 \fi  
1317 }%
```

Same code for the closing quote.

```
1318 \define@key{FB}{fg}{%  
1319     \ifFBunicode  
1320         \ifFB@luatex@punct  
1321             \FB@addGUILspace=1 \relax  
1322         \fi  
1323         \ifFB@xetex@punct  
1324             \XeTeXcharclass"14 = \FB@guilf  
1325             \XeTeXcharclass"BB = \FB@guilf  
1326             \XeTeXcharclass"A0 = \FB@guilnul  
1327             \XeTeXcharclass"202F = \FB@guilnul  
1328         \fi  
1329     \else  
1330         \newcommand*\FB@fg{%  
1331             \iflanguage{french}%  
1332                 {\ifFBAutoSpaceGuill\FB@fg  
1333                  \else\guillemotright  
1334                  \fi}%  
1335                 {\guillemotright}}%  
1336     \AtBeginDocument{%  
1337         \ifdefined\DeclareInputText  
1338             \ifdefined\uc@dclc  
1339                 \uc@dclc{187}{default}{\FB@fg}%  
1340             \else  
1341                 \ifdefined\DeclareUnicodeCharacter  
1342                     \DeclareUnicodeCharacter{00BB}{\FB@fg}%  
1343                 \else  
1344                     \@tempcnta'#1\relax  
1345                     \ifdefined\mule@def  
1346                         \mule@def{27}{\FB@fg}%  
1347                     \else  
1348                         \DeclareInputText{\the\@tempcnta}{\FB@fg}%  
1349                     \fi  
1350                 \fi  
1351             \fi  
1352         \else  
1353             \PackageWarning{frenchb.ldb}%  
1354             {Option 'fg' requires package inputenc.\MessageBreak}%  
1355         \fi  
1356     }%  
1357 \fi  
1358 }%  
1359 }
```

`\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`.

```
1360 \newcommand*{\FBprocess@options}{%
```

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

```
1361 \ifpackageloaded{enumitem}{%
1362   \ifFBStandardItemizeEnv
1363   \else
1364     \FBStandardItemizeEnvtrue
1365     \PackageInfo{frenchb.ldf}%
1366     {Setting StandardItemizeEnv=true for\MessageBreak
1367      compatibility with enumitem package,\MessageBreak}%
1368   \fi
1369   \ifFBStandardEnumerateEnv
1370   \else
1371     \FBStandardEnumerateEnvtrue
1372     \PackageInfo{frenchb.ldf}%
1373     {Setting StandardEnumerateEnv=true for\MessageBreak
1374      compatibility with enumitem package,\MessageBreak}%
1375   \fi}}%
1376 \ifpackageloaded{paralist}{%
1377   \ifFBStandardItemizeEnv
1378   \else
1379     \FBStandardItemizeEnvtrue
1380     \PackageInfo{frenchb.ldf}%
1381     {Setting StandardItemizeEnv=true for\MessageBreak
1382      compatibility with paralist package,\MessageBreak}%
1383   \fi
1384   \ifFBStandardEnumerateEnv
1385   \else
1386     \FBStandardEnumerateEnvtrue
1387     \PackageInfo{frenchb.ldf}%
1388     {Setting StandardEnumerateEnv=true for\MessageBreak
1389      compatibility with paralist package,\MessageBreak}%
1390   \fi}}%
1391 \ifpackageloaded{enumerate}{%
1392   \ifFBStandardEnumerateEnv
1393   \else
1394     \FBStandardEnumerateEnvtrue
1395     \PackageInfo{frenchb.ldf}%
1396     {Setting StandardEnumerateEnv=true for\MessageBreak
1397      compatibility with enumerate package,\MessageBreak}%
1398   \fi}}%
```

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

main language:

```
1399 \def\FB@ufl{\update@frenchlists}
1400 \ifx\bbl@main@language\FB@french
1401   \update@frenchlists
1402 \fi
```

The layout of footnotes is handled at the `\begin{document}` depending on the values of flags `FrenchFootnotes` and `AutoSpaceFootnotes` (see section 2.12), nothing has to be done here for footnotes.

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

```
1403 \ifBFAutoSpacePunctuation
1404   \autospace@beforeFDP
1405 \else
1406   \noautospace@beforeFDP
1407 \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.

```
1408 \ifFBOriginalTypewriter
1409 \else
1410   \let\ttfamilyORI\ttfamily
1411   \let\rmfamilyORI\rmfamily
1412   \let\sffamilyORI\sffamily
1413   \let\ttfamily\ttfamilyFB
1414   \let\rmfamily\rmfamilyFB
1415   \let\sffamily\sffamilyFB
1416 \fi
```

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

```
1417 \ifFBThinColonSpace
1418   \ifFB@luatex@punct
1419     \FBcolonskip=\FBthinspace\relax
1420   \else
1421     \renewcommand*\FBcolonspace{\FBthinspace}%
1422   \fi
1423 \fi
```

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

```
1424 \ifFBINGuillSpace
1425   \ifFB@luatex@punct
1426     \FBguillskip=3.33pt plus 1.665pt minus 1.11pt \relax
1427   \else
1428     \renewcommand*\FBguillspace{\space}%
1429   \fi
1430 \fi
```

When `true`, `ThinSpaceInFrenchNumbers` redefines `numprint.sty`’s command `\npstylefrench` to set `\npthousandsep` to `\`, (`thinspace`) instead of `~` (default)

. This option has no effect if package numprint is not loaded with ‘autolanguage’. As old versions of numprint did not define \npstylefrench, we have to provide this command.

```

1431 \ifpackageloaded{numprint}%
1432 {\ifnprt@autolanguage
1433   \providecommand*\npstylefrench{}}%
1434   \ifFBThinSpaceInFrenchNumbers
1435     \renewcommand*\npstylefrench{%
1436       \npthousandsep{\,}%
1437       \npdecimalsign{,}%
1438       \npproductsign{\cdot}%
1439       \npunitseparator{\,}%
1440       \npdegreeseperator{}}%
1441     \nppercentseparator{\nprt@unitsep}%
1442   }%
1443   \else
1444     \renewcommand*\npstylefrench{%
1445       \npthousandsep{~}%
1446       \npdecimalsign{,}%
1447       \npproductsign{\cdot}%
1448       \npunitseparator{\,}%
1449       \npdegreeseperator{}}%
1450     \nppercentseparator{\nprt@unitsep}%
1451   }%
1452   \fi
1453   \npaddtolanguage{french}{french}%
1454   \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.

```

1455 \ifFBFrenchSuperscripts
1456   \DeclareRobustCommand*\up{\ifstar{\FB@up@fake}{\fup}}%
1457   \else
1458     \DeclareRobustCommand*\up{\ifstar{\FB@up@fake}%
1459                                   {\textsuperscript}}%
1460   \fi

```

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

```

1461 \ifBFLowercaseSuperscripts
1462 \else
1463   \renewcommand*\FB@lc[1]{##1}%
1464 \fi

```

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

```

1465 \ifBFCustomiseFigTabCaptions
1466   \ifFB@koma
1467     \let\captionformat\CaptionSeparator

```

```

1468 \fi
1469 \@ifclassloaded{memoir}%
1470 {\captiondelim{\CaptionSeparator}}}%
1471 \else

```

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

```

1472 \ifFB@koma
1473 \def\captionformat{\iflanguage{french}{ : }{: }}%
1474 \fi
1475 \@ifclassloaded{memoir}%
1476 {\captiondelim{\iflanguage{french}{ : }{: }}}%
1477 \fi

```

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

```

1478 \ifBShowOptions
1479 \GenericWarning{* }{%
1480 * **** List of possible options for frenchb ****\MessageBreak
1481 [Default values between brackets when frenchb is loaded *LAST*]%
1482 \MessageBreak
1483 ShowOptions=true [false]\MessageBreak
1484 StandardLayout=true [false]\MessageBreak
1485 GlobalLayoutFrench=false [true]\MessageBreak
1486 StandardLists=true [false]\MessageBreak
1487 IndentFirst=false [true]\MessageBreak
1488 ReduceListSpacing=false [true]\MessageBreak
1489 ListOldLayout=true [false]\MessageBreak
1490 StandardItemizeEnv=true [false]\MessageBreak
1491 StandardEnumerateEnv=true [false]\MessageBreak
1492 StandardItemLabels=true [false]\MessageBreak
1493 ItemLabels=\textendash, \textbullet,
1494 \protect\ding{43},... [\textendash]\MessageBreak
1495 ItemLabeli=\textendash, \textbullet,
1496 \protect\ding{43},... [\textendash]\MessageBreak
1497 ItemLabelii=\textendash, \textbullet,
1498 \protect\ding{43},... [\textendash]\MessageBreak
1499 ItemLabeliii=\textendash, \textbullet,
1500 \protect\ding{43},... [\textendash]\MessageBreak
1501 ItemLabeliv=\textendash, \textbullet,
1502 \protect\ding{43},... [\textendash]\MessageBreak
1503 FrenchFootnotes=false [true]\MessageBreak
1504 AutoSpaceFootnotes=false [true]\MessageBreak
1505 AutoSpacePunctuation=false [true]\MessageBreak
1506 OriginalTypewriter=true [false]\MessageBreak
1507 ThinColonSpace=true [false]\MessageBreak
1508 ThinSpaceInFrenchNumbers=true [false]\MessageBreak
1509 FrenchSuperscripts=false [true]\MessageBreak
1510 LowercaseSuperscripts=false [true]\MessageBreak
1511 PartNameFull=false [true]\MessageBreak
1512 SuppressWarning=true [false]\MessageBreak
1513 CustomiseFigTabCaptions=false [true]\MessageBreak
1514 OldFigTabCaptions=true [false]\MessageBreak

```

```

1515 SmallCapsFigTabCaptions=false [true]\MessageBreak
1516 INGuilSpace=true [false]\MessageBreak
1517 InnerGuillSingle=true [false]\MessageBreak
1518 EveryParGuill=open, close, none [open]\MessageBreak
1519 EveryLineGuill=open, close, none
1520             [open in LuaTeX, none otherwise]\MessageBreak
1521 og= <left quote character>, fg= <right quote character>%
1522 \MessageBreak
1523 *****%
1524 \MessageBreak\protect\frenchbsetup{ShowOptions}}
1525 \fi
1526 }

```

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.

```

1527 \AtBeginDocument{%
1528   \providecommand*\xspace{\relax}%

```

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

```

1529   \ifdefined\pdfstringdefDisableCommands
1530     \pdfstringdefDisableCommands{%
1531       \let\up\relax
1532       \let\up\relax
1533       \let\degre\textdegree
1534       \let\degres\textdegree
1535       \def\ieme{e\xspace}%
1536       \def\iemes{es\xspace}%
1537       \def\ier{er\xspace}%
1538       \def\iers{ers\xspace}%
1539       \def\iere{re\xspace}%
1540       \def\ieres{res\xspace}%
1541       \def\FrenchEnumerate#1{#1\degre\space}%
1542       \def\FrenchPopularEnumerate#1{#1\degre)\space}%
1543       \def\No{N\degre\space}%
1544       \def\no{n\degre\space}%
1545       \def\Nos{N\degre\space}%
1546       \def\nos{n\degre\space}%
1547       \def\FB@og{\guillemotleft\space}%
1548       \def\FB@fg{\space\guillemotright}%
1549       \def\at{@}%
1550       \def\circonflexe{\string^}%
1551       \def\tild{\string~}%
1552       \let\bsc\textsc
1553     }%
1554   \fi

```

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

```

1555   \FBprocess@options

```


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

```
1556 \ifFB@luatex@punct
1557   \activate@luatexpunct
1558 \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!

```
1559 \ifFBunicode
1560   \ifdefined\DeclareUTFcharacter
1561   \else
1562     \@ifpackageloaded{luainputenc}{}%
1563     {\PackageWarning{frenchb.ldf}%
1564      {Add \protect\usepackage{fontspec} to the\MessageBreak
1565       preamble of your document,}%
1566     }%
1567   \fi
1568 \else
1569   \begingroup \newcommand{\FBOTone}{OT1}%
1570   \ifx\encodingdefault\FBOTone
1571     \PackageWarning{frenchb.ldf}%
1572     {OT1 encoding should not be used for French.%
1573      \MessageBreak
1574      Add \protect\usepackage[T1]{fontenc} to the
1575      preamble\MessageBreak of your document,}%
1576   \fi
1577   \endgroup
1578 \fi
1579 }
```

2.10 French lists

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

```
1580 \let\listORI\list
```

```

1581 \let\endlistORI\endlist
1582 \def\FB@listVsettings{%
1583     \setlength{\itemsep}{0.4ex plus 0.2ex minus 0.2ex}%
1584     \setlength{\parsep}{0.4ex plus 0.2ex minus 0.2ex}%
1585     \setlength{\topsep}{0.8ex plus 0.4ex minus 0.4ex}%
1586     \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.
1587     \@tempdima=\parskip
1588     \addtolength{\topsep}{-\@tempdima}%
1589     \addtolength{\partopsep}{\@tempdima}%
1590 }
1591 \def\listFB#1#2{\listORI{#1}{\FB@listVsettings #2}}
1592 \let\endlistFB\endlist

```

Let's now consider French itemize-lists. They differ from those provided by the standard L^AT_EX_{2 ϵ} classes:

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

<p style="text-align: center;">Text starting at 'parindent'</p> <p>← Leftmargin</p> <p>— first item. . .</p> <p style="padding-left: 2em;">— first second level item</p> <p style="padding-left: 2em;">— next one. . .</p> <p>— second item. . .</p>
--

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

```

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

```

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

```

1598 \newlength\listindentFB
1599 \setlength\listindentFB{-1pt}
1600 \newlength\labelwidthFB
1601 \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.

```

1602 \newlength\leftmarginFB
1603 \def\FB@listHsettings{%
1604   \leftmarginFB\labelwidthFB
1605   \advance\leftmarginFB \labelsep
1606   \leftmargini\leftmarginFB
1607   \advance\leftmargini \listindentFB
1608   \leftmarginii\leftmarginFB
1609   \leftmarginiii\leftmarginFB
1610   \leftmarginiv\leftmarginFB
1611   \leftmargin\csname leftmargin\romannumeral\the\@listdepth\endcsname
1612 }

```

`\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).

```

1613 \def\FB@itemizesettings{%
1614   \ifFBReduceListSpacing
1615     \setlength{\itemsep}{\z@}%
1616     \setlength{\parsep}{\z@}%
1617     \setlength{\topsep}{\z@}%
1618     \setlength{\partopsep}{\z@}%
1619     \@tempdima=\parskip
1620     \addtolength{\topsep}{-\@tempdima}%
1621     \addtolength{\partopsep}{\@tempdima}%
1622   \fi
1623   \settowidth{\labelwidth}{\csname\@itemitem\endcsname}%
1624   \ifFBListOldLayout
1625     \setlength{\leftmargin}{\labelwidth}%
1626     \addtolength{\leftmargin}{\labelsep}%
1627     \addtolength{\leftmargin}{\parindent}%
1628   \else
1629     \FB@listHsettings
1630   \fi
1631 }

```

The definition of `\itemizeFB` follows the one of `\itemize` in standard L^AT_EX_{2 ϵ} classes (see `ltxlists.dtx`), spaces are customised by `\FB@itemizesettings`.

```

1632 \def\itemizeFB{%
1633   \ifnum \@itemdepth >\thr@@\@toodeep\else
1634     \advance\@itemdepth\@ne
1635     \edef\@itemitem{\labelitem\romannumeral\the\@itemdepth}%
1636     \expandafter
1637     \listORI
1638     \csname\@itemitem\endcsname
1639     \FB@itemizesettings

```

```

1640   \fi
1641 }
1642 \let\enditemizeFB\endlistORI

1643 \def\labelitemsFB{%
1644   \let\labelitemi\Frlabelitemi
1645   \let\labelitemii\Frlabelitemii
1646   \let\labelitemiii\Frlabelitemiii
1647   \let\labelitemiv\Frlabelitemiv
1648   \ifdim\labelwidthFB<\z@
1649     \settowidth{\labelwidthFB}{\FrenchLabelItem}%
1650   \fi
1651   \ifdim\listindentFB<\z@
1652     \ifdim\parindent=\z@
1653       \setlength{\listindentFB}{1.5em}%
1654     \else
1655       \setlength{\listindentFB}{\parindent}%
1656     \fi
1657   \fi
1658 }

```

\enumerateFB The definition of `\enumerateFB`, new to version 2.6a, follows the one of `\enumerate` in standard L^AT_εX classes (see `ltlists.dtx`), vertical spaces are customised (or not) via `\list` (`=\listFB` or `\listORI`) and horizontal spaces (leftmargins) are borrowed from `itemize` lists via `\FB@listHsettings`.

```

1659 \def\enumerateFB{%
1660   \ifnum \@enumdepth >\thr@@\@toodeep\else
1661     \advance\@enumdepth\@ne
1662     \edef\@enumctr{enum\romannumeral\the\@enumdepth}%
1663     \expandafter
1664     \list
1665       \csname label\@enumctr\endcsname
1666       {\FB@listHsettings
1667         \usecounter\@enumctr\def\makelabel##1{\hss\llap{##1}}}%
1668   \fi
1669 }
1670 \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).

```

1671 \def\descriptionFB{%
1672   \list{}{\FB@listHsettings
1673     \labelwidth\z@
1674     \itemindent-\leftmargin
1675     \ifnum\@listdepth=1
1676       \advance\itemindent by \listindentFB
1677     \fi
1678     \let\makelabel\descriptionlabel}%
1679 }
1680 \let\enddescriptionFB\endlistORI

```

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

```
\bbl@frenchlistlayout 1681 \def\update@frenchlists{%
\bbl@nonfrenchlistlayout 1682 \ifFBReduceListSpacing \let\list\listFB \fi
1683 \ifFBStandardItemizeEnv
1684 \else \let\itemize\itemizeFB \fi
1685 \ifFBStandardItemLabels
1686 \else \labelitemsFB \fi
1687 \ifFBStandardEnumerateEnv
1688 \else \let\enumerate\enumerateFB \let\description\descriptionFB \fi
1689 }
```

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.

```
1690 \def\FB@ufl{\relax}
1691 \def\bbl@frenchlistlayout{%
1692 \ifFBGlobalLayoutFrench
1693 \else
1694 \babel@save\list \babel@save\itemize
1695 \babel@save\enumerate \babel@save\description
1696 \babel@save\labelitemi \babel@save\labelitemii
1697 \babel@save\labelitemiii \babel@save\labelitemiv
1698 \fi
1699 \FB@ufl
1700 }
1701 \def\bbl@nonfrenchlistlayout{%
1702 \ifFBGlobalLayoutFrench
1703 \update@frenchlists
1704 \fi
1705 }
1706 \FB@addto{extras}{\bbl@frenchlistlayout}
1707 \FB@addto{noextras}{\bbl@nonfrenchlistlayout}
```

2.11 French indentation of sections

`\bbl@frenchindent` In French the first paragraph of each section should be indented, this is another difference with US-English. This is controlled by the flag `\if@afterindent`.

`\bbl@nonfrenchindent` We will need to save the value of the flag `\if@afterindent` ‘AtBeginDocument’ before eventually changing its value.

```
1708 \def\bbl@frenchindent{%
1709 \ifFBGlobalLayoutFrench\else\babel@save\@afterindentfalse\fi
1710 \ifFBIndentFirst
1711 \let\@afterindentfalse\@afterindenttrue
1712 \@afterindenttrue
1713 \fi}
1714 \def\bbl@nonfrenchindent{%
1715 \ifFBGlobalLayoutFrench
1716 \ifFBIndentFirst
1717 \@afterindenttrue
```

```

1718     \fi
1719 \fi}
1720 \FB@addto{extras}{\bbl@frenchindent}
1721 \FB@addto{noextras}{\bbl@nonfrenchindent}

```

2.12 Formatting footnotes

The `bigfoot` package deeply changes the way footnotes are handled. When `bigfoot` is loaded, we just warn the user that `frenchb` will drop the customisation of footnotes. The layout of footnotes is controlled by two flags `\ifFBAutoSpaceFootnotes` and `\ifBFFrenchFootnotes` which are set by options of `\frenchbsetup{}` (see section 2.9). The layout of footnotes *does not depend* on the current language (just think of two footnotes on the same page looking different because one was called in a French part, the other one in English!).

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

```

1722 \AtBeginDocument{\@ifpackageloaded{bigfoot}%
1723     {\PackageInfo{frenchb.ldf}%
1724     {bigfoot package in use.\MessageBreak
1725     frenchb will NOT customise footnotes;\MessageBreak
1726     reported}}%
1727     {\let\@footnotemarkORI\@footnotemark
1728     \def\@footnotemarkFB{\leavevmode\unskip\unkern
1729     \,\@footnotemarkORI}%
1730     \ifFBAutoSpaceFootnotes
1731     \let\@footnotemark\@footnotemarkFB
1732     \fi}%
1733     }

```

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).

```

1734 \newcommand*{\dotFFN}{.}
1735 \newcommand*{\kernFFN}{\kern .5em}
1736 \newdimen\parindentFFN
1737 \parindentFFN=10in
1738 \def\ftnISSymbol{\@fnsymbol\c@footnote}
1739 \long\def\@makefntextFB#1{\ifx\thefootnote\ftnISSymbol
1740     \@makefntextORI{#1}%

```

```

1741         \else
1742         \parindent=\parindentFFN
1743         \rule\z@\footnotesep
1744         \setbox\@tempboxa\hbox{\@thefnmark}%
1745         \ifdim\wd\@tempboxa>\z@
1746         \llap{\@thefnmark}\dotFFN\kernFFN
1747         \fi #1
1748     \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).

```

1749 \AtBeginDocument{\@ifpackageloaded{bigfoot}{}%
1750     {\ifdim\parindentFFN<10in
1751         \else
1752         \parindentFFN=\parindent
1753         \ifdim\parindentFFN<1.5em \parindentFFN=1.5em \fi
1754         \fi
1755         \let\@makefnORI\@makefn
1756         \long\def\@makefn#1{%
1757             \ifFBFrenchFootnotes
1758             \@makefnFB{#1}%
1759             \else
1760             \@makefnORI{#1}%
1761             \fi}%
1762     }%
1763 }

```

For compatibility reasons, we provide definitions for the commands dealing with the layout of footnotes in frenchb version 1.6. `\frenchbsetup{}` (see in section 2.9) should be preferred for setting these options. `\StandardFootnotes` may still be used locally (in minipages for instance), that's why the test `\ifFBFrenchFootnotes` is done inside `\@makefn`.

```

1764 \newcommand*\AddThinSpaceBeforeFootnotes{\FBAutoSpaceFootnotestru}
1765 \newcommand*\FrenchFootnotes{\FBFrenchFootnotestru}
1766 \newcommand*\StandardFootnotes{\FBFrenchFootnotesfalse}

```

2.13 Clean up and exit

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

```

1767 \FBclean@on@exit
1768 \ldf@finish\CurrentOption

```

3 Change History

v2.0	General: <code>\parindentFFN</code> not changed if already defined (required by JA for <code>cah-gut.cls</code>).	62		
	Added warning for OT1 encoding.	56		
	Footnotes are now printed by default ‘à la française’ for the whole document.	62		
	New command <code>\frenchbsetup</code> added for global customisation.	45		
	<code>\bsc</code> : <code>\hbox</code> dropped, replaced by <code>\kern0pt</code>	37		
	<code>\captionsfrench</code> : ‘Fig.’ changed to ‘Figure’ and ‘Tab.’ to ‘Table’.	40		
	<code>\datefrench</code> : 2 ‘relax’ added in <code>\today</code> ’s definition.	34		
	<code>\FBtextellipsis</code> : Added special case for LY1 encoding, see bug report from Bruno Voisin (2004/05/18).	44		
	<code>\nombre</code> : <code>\nombre</code> now requires <code>numprint.sty</code>	39		
v2.0b	General: Footnotes: Just do nothing (except warning) when the <code>bigfoot</code> package is loaded.	62		
v2.0c	General: There is no need to define here <code>numprint</code> ’s command <code>\npstylefrench</code> , it will be redefined ‘AtBeginDocument’ by <code>\FBprocess@options</code>	40		
	<code>\frenchbsetup</code> : Option <code>ThinSpaceInFrenchNumbers</code> added.	45		
v2.0d	<code>\frenchbsetup</code> : Options <code>og</code> and <code>fg</code> changed: limit the definition to French so that quote characters can be used in German.	45		
v2.0e	<code>\frenchbsetup</code> : New option: <code>StandardLists</code>	45		
v2.0f	<code>\frenchbsetup</code> : <code>StandardLayout</code> option had no effect on lists. Test moved to <code>\FBprocess@options</code>	45		
	Two typos corrected in option <code>StandardLists</code> : <code>[false]</code> → <code>[true]</code> to			
	and <code>StandardLayout</code> → <code>StandardLists</code>	45		
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>	45		
v2.1a	General: Command <code>\fup</code> added to produce better superscripts than <code>\textsuperscript</code>	34		
	<code>\datefrench</code> : <code>\today</code> changed (correction in 2.0 was wrong: <code>\today</code> was printed without spaces in <code>toc</code>).	34		
	<code>\frenchbsetup</code> : New option: <code>FrenchSuperscripts</code> to define <code>\up</code> as <code>\fup</code> or as <code>\textsuperscript</code>	45		
	New option: <code>LowercaseSuperscripts</code>	45		
v2.1b	General: Disable some commands in bookmarks.	56		
	<code>\fup</code> : Command <code>\fup</code> changed to use real superscripts from <code>fourier v. 1.6</code>	35		
v2.1c	General: Added commands <code>\Nos</code> and <code>\nos</code>	36		
	<code>\degres</code> : Provide a temporary definition (hyperref safe) of <code>\degres</code> in case it has to be expanded in the preamble (by beamer’s <code>\title</code> command for instance).	38		
	<code>\up</code> : Provide a temporary definition (hyperref safe) of <code>\up</code> in case it has to be expanded in the preamble (by beamer’s <code>\title</code> command for instance).	35		
v2.1d	General: Argument of <code>\ProvidesLanguage</code> changed above from ‘french’ to			

‘frenchb’ (otherwise <code>\listfiles</code> prints no date/version information). The real name of current language (french) as to be corrected before calling <code>\LdfInit</code> . 13	(suggested by JA). 62
Avoid warning “\end occurred when <code>\ifx ... incomplete</code> ” with LaTeX-2.09. 13	
v2.2a	v2.3c
<code>\frenchbsetup</code> : Default values of flags changed: default now means ‘StandardLayout’, they will be changed to ‘FrenchLayout’ AtEndOfPackage only if french is <code>\bbl@main@language</code> 45	General: Commands <code>\ttfamily</code> , <code>\rmfamily</code> and <code>\sffamily</code> have to be robust. Bug introduced in 2.3a, pointed out by Manuel Pégourié-Gonnard. 29
The global layout of the document is no longer changed when frenchb is not the last option of babel (<code>\bbl@main@language</code>). Suggested by Ulrike Fischer. 45	v2.3d
When frenchb is babel’s last option, French becomes the document’s main language, so GlobalLayout-French applies. 45	<code>\bbl@nonfrenchindent</code> : Bug correction: previous versions of frenchb set the flag <code>\if@afterindent</code> to false outside French which is correct for English but wrong for some languages like Spanish. Pointed out by Juan José Torrens. 61
<code>\fup</code> : <code>\newif</code> and <code>\newdimen</code> moved before <code>\ifLaTeXe</code> to avoid an error with plainTeX. 35	v2.3e
v2.3a	General: Execute <code>\AutoSpaceBeforeFDP</code> also in LaTeX to define <code>\FDP@colonspace</code> : needed for tex4ht, pointed out by MPG. . . . 28
General: <code>\NoAutoSpaceBeforeFDP</code> and <code>\AutoSpaceBeforeFDP</code> now set the flag <code>\ifFBAutoSpacePunctuation</code> accordingly (LaTeX only). 28	v2.4a
In LaTeX, frenchb no longer adds spaces before ‘high punctuation’ characters in computer code. Suggested by Yannis Haralambous. 29	General: <code>\PackageWarning</code> changed to <code>\FBWarning</code> (when bigfoot package in use). 62
<code>\frenchbsetup</code> : New option: <code>OriginalTypewriter</code> . Now frenchb switches to <code>\noautospace@beforeFDP</code> when a tt-font is in use. When <code>OriginalTypewriter</code> is set to true, frenchb behaves as in pre-2.3 versions. 45	<code>\CaptionSeparator</code> : <code>\PackageWarning</code> changed to <code>\FBWarning</code> (in case <code>\@makecaption</code> has been customised). <code>\FBWarning</code> is defined as <code>\PackageWarning</code> by default but can be made silent using <code>\frenchbsetup</code> , (suggested by MPG). 42
<code>\fup</code> : <code>\lowercase</code> changed to <code>\MakeLowercase</code> as the former doesn’t work for non ASCII characters in encodings like applemac, utf-8, 35	<code>\frenchbsetup</code> : New option <code>SuppressWarning</code> 45
v2.3b	<code>\ifFBXeTeX</code> : Added a new ‘if’ <code>\FBunicode</code> and some <code>\lccode</code> definitions to <code>\extrasfrench</code> and <code>\noextrasfrench</code> 15
General: New commands <code>\dotFFN</code> and <code>\kernFFN</code> for more flexibility	v2.4c
	General: In <code>\ttfamilyFB</code> , also cancel automatic spaces inside French guillemets coded as characters (see <code>\frenchbsetup</code>). 49
	In <code>\ttfamilyFB</code> , also cancel automatic spaces inside French guillemets entered as characters (see <code>\frenchbsetup</code>). 29
	v2.4d
	<code>\up</code> : Command <code>\up</code> defined with <code>\providecommand</code> instead of

	<code>\newcommand</code> as <code>\up</code> may be defined elsewhere (<code>catalan.ldf</code>). Bug pointed out by Felip Manyé i Ballester.	35		
v2.5a	General: <code>\og</code> and <code>\fg</code> do not print correctly in English when using XeTeX or LuaTeX, fixed by using <code>\textquotedblleft</code> and <code>\textquotedblright</code> defined above.	31		
	New command <code>\NoAutoSpacing</code> , suggested by MPG.	29		
	Punctuation is no longer made active with XeTeX-based engines.	16		
	<code>\captionsfrench</code> : <code>\emph</code> deleted in <code>\seename</code> and <code>\alsoname</code> to match what is done for the other languages. Suggested by Marc Baudoin.	40		
	<code>\FBthinspace</code> : Define <code>\FBthinspace</code> for those who want to customise the width of the space before ; and co.	17		
	<code>\textquoteddblright</code> : Change <code>\guillemotleft</code> and <code>\guillemotright</code> definitions for Unicode and provide definitions for <code>\textquotedblleft</code> and <code>\textquotedblright</code> . Insures correct printing of quotes by <code>\og</code> and <code>\fg</code> in French and outside.	30		
v2.5b	General: Do not use the test <code>\iflanguage{french}</code> to check whether French is the main language or not, as it might be erroneously positive when English is the main language and no hyphenation patterns are available for French. In this case <code>\l@french</code> and <code>\l@english</code> are 0. Pointed out by Günter Milde.	46		
v2.5c	General: The code meant for XeTeX also works for LuaTeX, we just need to change the test.	49		
v2.5d	General: Moved the <code>\newcount</code> command outside <code>\ifFB@xetex@punct</code> ... <code>\fi</code> (it broke Plain formats). .	24		
				<code>\ifFBXeTeX</code> : Added two new ‘if’ <code>\FBXeTeX</code> and <code>\FBLuaTeX</code> as XeTeX and behave differently regarding the status of the French “apostrophe”.
				15
v2.5e	General: <code>\pdfstringdefDisableCommands</code> should redefine <code>\FB@og</code> and <code>\FB@fg</code> instead of <code>\og</code> and <code>\fg</code> so that it works also when quotes are entered as characters. Reported by Sébastien Gouezel.	56		
v2.5f	General: Changed definitions of <code>\at</code> , <code>\circflexe</code> , <code>\tild</code> , <code>\boi</code> and <code>\degre</code> for Unicode based engines.	37		
	<code>\FBtextellipsis</code> : Unicode fonts also provide a ready made character for <code>\textellipsis</code> , let’s just use it (reported by Maxime Chupin, 2011/06/04).	44		
v2.5g	General: Redefine <code>\degre</code> , <code>\degres</code> <code>\at</code> <code>\circflexe</code> and <code>\tild</code> for bookmarks. Add <code>\fup</code> also.	56		
	When <code>\ifFB@xetex@punct</code> is true, ‘og’ and ‘fg’ options now set XeTeXcharclasses of these characters to <code>\FB@guilo</code> and <code>\FB@guilf</code> . Otherwise French quotes behave as normal characters (their XeTeXcharclass is 0).	49		
	<code>\FB@xetex@punct@french</code> : XeTeXcharclass(es) for French quotes will be set to <code>\FB@guilo</code> and <code>\FB@guilf</code> by options ‘og’ and ‘fg’ in <code>\frenchbsetup</code> . French quotes should behave as normal characters by default in XeLaTeX as in LaTeX.	25		
v2.5h	<code>\degres</code> : <code>textcomp.sty</code> has changed. The test about <code>\M@TS1</code> is no longer relevant, let’s change it.	38		
v2.5i	General: Temporary fix: as long as <code>xeCJK.sty</code> will not use <code>\newXeTeXintercharclass</code> to allocate its classes, we will have to define 3 fake classes.	24		

<code>\FB@xetex@punct@french: xCJK.sty</code> changes the <code>\XeTeXcharclass</code> of ASCII chars ‘-’ ‘ ’ ‘ ’ ‘ ’ ‘ ’ ‘ ’ ‘ ’ ‘{’ ‘%’ opening and closing single and double quotes. We set their class to 0 in French and re- set their class to their original value when leaving French. See <code>\FB@xetex@punct@nonfrench</code> be- low. 25	<code>Newif \if@FBwarning@capsep</code> added. 42
v2.5j General: Previous fix removed: bug fixed in <code>xCJK.sty</code> version 3.0.4 (06-May-2012). 24	v2.6d <code>\FBthinspace</code> : Rename <code>\Fthinspace</code> to <code>\FBthinspace</code> and <code>\Fcolonspace</code> to <code>\FBcolonspace</code> to avoid a con- flict with <code>fournier.sty</code> 17
v2.6a General: Bug correction: changing <code>\leftmargin</code> cannot be done only for <code>itemize</code> -lists: it messes up em- bedded <code>enumerate</code> lists. Pointed out by Denis Bitouzé. Lists have been completely redesigned in <code>frenchb v. 2.6a</code> . An option for backward compatibility is pro- vided. 57	v2.6e <code>\degrees</code> : Refrain from redefining <code>\textdegree</code> from <code>latin1.def</code> , <code>ap- plemac.def</code> , etc. as <code>\degrees</code> be- cause it loops in <code>hyperref</code> ’s book- marks. Pointed out by Eddy Flas on <code>fctt</code> 38
<code>\frenchbsetup</code> : New options <code>ListOld-</code> <code>Layout</code> , <code>StandardItemizeEnv</code> and <code>StandardEnumerateEnv</code> (Com- <code>compactItemize</code> is deprecated). 45	v2.6f <code>\FB@itemizesettings: \labelwidth</code> must be reset, f.i. when an <code>itemize</code> list occurs inside environ- ments based on <code>trivlist</code> which set <code>\labelwidth</code> to 0 (see proof en- vironment in <code>amsthm.sty</code>). Bug pointed out by Julien Hauseux. . 59
<code>\FrenchLabelItem</code> : default changed from <code>\textendash</code> to <code>\textemdash</code> 58	v2.6g General: U+00A0 (Unicode no- breakspace) and U+202F (Uni- code nobreakthinspace) added to class <code>\FB@punctnul</code> to prevent <code>frenchb</code> from adding it’s own space before ‘high punctuation’ characters. 24
v2.6b <code>\descriptionFB</code> : Settings of <code>\FB@listHsettings</code> should apply to <code>description</code> lists too. 60	<code>\FB@itemizesettings</code> : Suppress all vertical spaces only if <code>Re- duceListSpacing</code> is true. Pointed out by Pierre Willaime. 59
v2.6c General: Dummy file <code>frenchb.cfg</code> is no longer generated from <code>frenchb.dtx</code> 12	<code>\ifFBXeTeX</code> : <code>lccode</code> values for the French “apostrophe” are now the same for <code>XeTeX</code> and <code>LuaTeX</code> 15
No warning about <code>\@makecaption</code> for <code>AMS</code> classes. 42	v2.6h General: <code>\FG@og</code> and <code>\FG@fg</code> changed: former clumsy code removed. .. 49
No warning about <code>\@makecaption</code> for <code>koma-script</code> classes. <code>\captionformat</code> customised in French. 43	If <code>\@makecaption</code> is undefined, no warning. 43
Warning added when the <code>caption</code> or <code>floatrow</code> package is loaded before <code>babel/frenchb</code> 43	New class <code>\FB@guilnul</code> for char- acters U+00A0 (Unicode no- breakspace) and U+202F (Uni- code nobreakthinspace), to pre- vent <code>frenchb</code> from adding spurious spaces inside quotes. 24
<code>\CaptionSeparator</code> : Former <code>\CaptionSeparator</code> has been re- named as <code>\FBCaption@Separator</code> ;	<code>\CaptionSeparator</code> : No active cat- codes in <code>\STD@makecaption</code> ’s def- inition. 42

v3.0a	
General: <code>\LdfInit</code> checks	<code>\datefrench</code> . Doesn't work with Plain (yet?). 34
<code>\datefrench</code> instead of	<code>\descriptionFB</code> : Add <code>\listindentFB</code> to <code>\itemindent</code> . Suggested by Denis Bitouzé. 60
<code>\captionsfrench</code> to avoid a conflict with <code>papertex.cls</code> which loads <code>datetime.sty</code> 13	<code>\extrasfrench</code> : Take advantage of babel's <code>\babel@savevariable</code> to handle apostrophe's <code>\lccode</code> . . . 15
<code>\bbl@nonfrenchguillemets</code> deleted, use <code>\babel@save</code> instead. 31	<code>\FBprocess@options</code> : Changed option <code>ThinColonSpace</code> to make it work also with LuaTeX. 53
Added explicit <code>\FBguillskip</code> for LuaTeX. 31	With koma-script and memoir class, customise <code>\captionformat</code> and <code>\captiondelim</code> 54
Definitions of <code>\FB@og</code> and <code>\FB@fg</code> now depend on punctuation handling (LuaTeX / XeTeX / active). 31	<code>\FBthinspace</code> : LuaTeX requires dimensions: two new skips <code>\FBcolonskip</code> and <code>\FBthinspace</code> . 17
<code>french.cfg</code> will be loaded (if found) instead of <code>frenchb.cfg</code> . NO NEED for <code>.cfg</code> files in French anyway. . 63	<code>\FBthinspace</code> : <code>\FBthinspace</code> is no longer a kern but a skip (<code>frenchb</code> adds a <code>nobreak</code> penalty before it). 17
In Plain, provide a substitute for <code>\PackageWarning</code> and <code>\PackageInfo</code> 14	<code>\frenchbsetup</code> : New options <code>OldFigTabCaptions</code> and <code>CustomiseFigTabCaptions</code> 45
Merging of <code>\captionsfrenchb</code> , <code>\captionsfrançais</code> with <code>\captionsfrench</code> deleted in favor of new babel 3.9 syntax. . . . 41	v3.0b
More informative, less TeXnical warning about <code>@makecaption</code> . . 43	General: <code>frenchb.lua</code> was not found by Lua function <code>dofile</code> (not <code>kpathsea</code> aware). Call function <code>kpse.find_file</code> first, as suggested by Paul Gaborit. 23
New flag <code>\ifFB@luatex@punct</code> for 'high punctuation' management with LuaTeX engines. 16	Require <code>luatexbase</code> with <code>LaTeXe</code> in case <code>fontspec</code> has not been loaded before babel. 17
New handling of 'high punctuation' through callbacks with LuaTeX engines. 17	v3.0c
No warning about <code>@makecaption</code> for SMF classes. No warning either with LuaTeX or XeTeX engines. 42	General: Activate option <code>StandardLists</code> when <code>beamer</code> class is loaded. 46
Options processing completely reorganised. 45	Bug correction: null glues should not trigger space insertion before high punctuation. Bug pointed out by Benoit Rivet for the 'lstlisting' environment of the <code>listings</code> package. 20
Support for options <code>frenchb</code> , <code>français</code> , <code>canadien</code> , <code>acadian</code> changed. 13	Changed <code>\FBguill@spacing</code> (internal) to <code>\FBguillspace</code> (public). 31
Test <code>\ifXeTeX</code> changed to <code>\ifFBunicode</code> and 'xltextra' changed to 'fontspec'. 57	<code>frenchb</code> requires babel-3.9i. 14
<code>\CaptionSeparator</code> : Remove <code>\CaptionSeparatorORI</code> , use <code>\babel@save</code> instead. 42	Just load <code>luatexbase.sty</code> instead of <code>luaotfload.sty</code> with plain formats. 17
<code>\captionsfrench</code> : Take advantage of babel's <code>\SetString</code> commands for <code>captionnames</code> 40	Missing pair of braces after 'and' in the definition of 'auto'. 20
<code>\datefrench</code> : Take advantage of babel's <code>\SetString</code> commands for	No need to define <code>\l@french</code> as <code>\lang@french</code> , <code>babel.def</code> (3.9j)

takes care for this.	13		
\datefrench: \SetString still does not work for Plain with babel 3.9k. Need to define \datefrench. ..	34		
\frenchbsetup: New option IN- GuillSpace.	45		
v3.1a		v3.1b	
General: Codes "13 and "14 added for French quotes in T1-encoding. Support for older versions of Lua- TeX and XeTeX dropped.	49	General: frenchb.lua: add a check for null fid in french_punctuation (Tikz \nullfont). Bug pointed out by Paul Gaborit.	20
fontspec is not required for T1 fonts used with the luainputenc.sty package.	57	\captionsfrench: Change \scshape to customisable \FBfigtabshape for \figurename and \tablename.	40
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	\fprimo): Removed \lowercase from definitions of \FrenchEnumerate, ... \No and co: \up already does the conversion.	37
frenchb.lua: codes 0x13 and 0x14 added for French quotes in T1- encoding.	18	\frenchbsetup: New option Small- CapsFigTabCaptions.	45
frenchb.lua: look ahead when next is a kern (i.e in « \texttt{a} »).	21	\ieres: Removed \lowercase from definitions of \ieme and co: \up already does the conversion. ...	36
Misplaced \fi for plain formats. .	17		