

A Babel language definition file for French

frenchb.dtx v3.1l, 2016/02/13

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>	4
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	24
2.2.3 Punctuation with standard (pdf)TeX	27
2.2.4 Punctuation switches common to all engines	29
2.3 Commands for French quotation marks	30
2.4 Date in French	34
2.5 Extra utilities	35
2.6 Formatting numbers	39
2.7 Caption names	41
2.8 Dots...	45
2.9 More checks about packages' loading order	46
2.10 Setup options: keyval stuff	46
2.11 French lists	59
2.12 French indentation of sections	63
2.13 Formatting footnotes	63
2.14 Clean up and exit	65
3 Change History	66

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 v3.11 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.

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

¹The file described in this section has version number v3.11 and was last revised on 2016/02/13.

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).

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

1. French hyphenation patterns are made active;
2. ‘high punctuation’ characters (: ; ! ?) automatically add correct spacing in French; this is achieved using callbacks in Lua(La)TeX or ‘XeTeXinterchar’ mechanism in Xe(La)TeX; with TeX⁸² 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 opening or closing guillemet (« or ») depending on option `EveryLineGuill=open` (default) or `=close` unless you explicitly set `EveryLineGuill=none`, then `\frquote{}` will behave as with non-LuaTeX engines;
- with all other engines, the inner quotation is surrounded by double quotes (“*texte*”) unless option `InnerGuillSingle=true`, then a) the inner quotation is printed as `< texte >` and b) if the inner quotation spreads over more than one paragraph, every paragraph included in the inner quotation starts with a `<` or a `>`, depending on option `EveryParGuill=open` or `close`.

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

2. A command `\up` is provided to typeset superscripts like `M\up{me}` (abbreviation for “Madame”), `1\up{er}` (for “premier”). Other commands are also provided for ordinals: `\ier`, `\iere`, `\iers`, `\ieres`, `\ieme`, `\iemes` (`3\iemes` prints 3^{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{Lamport}` will print the same as `L.\mbox{\textsc{Lamport}}`. Note that composed names (such as Dupont-Durant) may now be hyphenated on explicit hyphens, this differs from frenchb v. 1.x.
4. Commands `\primo`, `\secundo`, `\tertio` and `\quarto` print 1^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 nobreak space), or for alcohols’ strengths (e.g., “45\degres” with *no* space in French).
7. In math mode the comma has to be surrounded with braces to avoid a spurious space being inserted after it, in decimal numbers for instance (see the T_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 ‘`1\ier juin`’ will print ‘1^{er} juin’ (no need for a forced space after `1\ier`).

1.2 Customisation

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

1.2.1 `\frenchbsetup{options}`

`\frenchbsetup{ShowOptions}` prints all available options to the .log file, it is just meant as a remainder of the list of offered options. As usual with `keyval`

syntax, boolean options (as `ShowOptions`) can be entered as `ShowOptions=true` or just `ShowOptions`, the `=true` part can be omitted.

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

`StandardLayout=true (false*)` forces `frenchb` not to interfere with the layout: no action on any kind of lists, first paragraphs of sections are not indented (as in English), no action on footnotes. This option can be used to avoid conflicts with classes or packages which customise lists or footnotes.

`GlobalLayoutFrench=false (true*)` should no longer be used; it was intended to emulate, when French is the main language, what prior versions of `frenchb` (pre-2.2) did: lists, and first paragraphs of sections would be displayed the standard way in other languages than French, and “à la française” in French. Note that the layout of footnotes is language independent anyway (see below `FrenchFootnotes` and `AutoSpaceFootnotes`).

`ReduceListSpacing=false (true*)` ; `frenchb` reduces the values of the vertical spaces used in the *all* list environments in French (this includes `itemize`, `enumerate`, `description`, but also `abstract`, `quote`, `quotation` and `verse` and possibly others). Setting this option to `false` reverts to the standard settings of the list environment.

`ListOldLayout=true (false)` ; starting with version 2.6a, the layout of lists has changed regarding leftmargins' sizes and default `itemize` label (`'—'` instead of `'-'` up to 2.5k). This option, provided for backward compatibility, displays lists as they were up to version 2.5k.

`CompactItemize=false (true*)` ; should no longer be used (kept only for backward compatibility), it is replaced by the next two options.

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

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

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

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

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

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

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

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

`StandardLists=true (false*)` forbids frenchb to customise any kind of list.

Try the option `StandardLists` in case of conflicts with classes or packages that customise lists too. This option is just a shorthand setting all four options `ReduceListSpacing=false`, `StandardItemizeEnv=true`, `StandardEnumerateEnv=true` and `StandardItemLabels=true`.

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

`FrenchFootnotes=false (true*)` reverts to the standard layout of footnotes. By default frenchb typesets leading numbers as ‘1. ’ instead of ‘¹’, but has no effect on footnotes numbered with symbols (as in the `\thanks` command). Two commands `\StandardFootnotes` and `\FrenchFootnotes` are available to change the layout of footnotes locally; `\StandardFootnotes` can help when some footnotes are numbered with letters (inside minipages for instance).

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

`FrenchSuperscripts=false (true)` ; then `\up=\textsuperscript`. (option added in version 2.1). Should only be made `false` to recompile documents written before 2008 without changes: by default `\up` now relies on `\fup` designed to produce better looking superscripts.

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

Choosing `AutoSpacePunctuation=false` will ensure that a proper space will be added before ‘:;!?’ *if and only if* a (normal) space has been typed in. Those who are unsure about their typing in this area should

stick to the default option and use the provided `\NoAutoSpacing` command inside a group in case an unwanted space is added by frenchb (i.e. `{\NoAutoSpacing 10:55}`).

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

1.2.2 Captions

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

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

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

Three new options are provided: if `CustomiseFigTabCaptions` is set to `false` the colon will be used as separator in all languages, with a proper space before the colon in French. The second option, `OldFigTabCaptions`, can be set to `true` to print figures’ and tables’ captions as they were with versions pre 3.0 of frenchb (using `\CaptionSeparator` in French and colon in other languages); this option only makes sense with the standard L^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” rather than in small caps (the default).

1.3 Hyphenation checks

Once you have built your format, a good precaution would be to perform some basic tests about hyphenation in French. For $\text{\LaTeX} 2_{\epsilon}$ I suggest this:

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

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

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

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

Frequent mismatches:

- you get sig-nal con-tainer, this probably means that the hyphenation patterns you are using are for US-English, not for French;
- you get no hyphen at all in évé-ne-ment, this probably means that you are using CM fonts and the macro `\accent` to produce accented characters. Using 8-bits fonts with built-in accented characters avoids this kind of mismatch.

1.4 Changes

What's new in version 3.1?

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

What's new in version 3.0?

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

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

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

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

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

What's new in version 2.6?

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

frenchb is now compatible with the paralist package.

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

What's new in version 2.5?

The main change is that active characters are no longer used in French with (recent) XeTeX-based engines (they still are with TeX-based engines). All the

functionalities (automatic insertion of missing spaces before `;!?` or bare replacement of typed spaces with suitable unbreakable ones, tuning of the spaces width) remain available and the user interface is unchanged. The use of active characters is replaced by the `\XeTeXinterchartoks` mechanism (adapted from the `polyglossia` package).

A new command `\NoAutoSpacing` has been added. It should be used *inside a group* instead of `\shorthandoff{;!?}` whenever active characters or automatic spacing of French punctuation or quote characters conflict with other packages; it is designed to work with TeX-, LuaTeX- and XeTeX-based engines.

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

What's new in version 2.4?

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

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

Better compatibility with the `enumitem` package.

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

What's new in version 2.3?

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

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

What's new in version 2.2?

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

What's new in version 2.1?

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

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

What's new in version 2.0?

Here is the list of all changes:

- Support for \LaTeX-2.09 and for $\text{\LaTeX2}_{\epsilon}$ in compatibility mode has been dropped. This version is meant for $\text{\LaTeX2}_{\epsilon}$ and Plain based formats (like `bplain`). $\text{\LaTeX2}_{\epsilon}$ formats based on `\text{\LaTeX}` are no longer supported either (plenty of good 8-bits fonts are available now, so T1 encoding should be preferred for typesetting in French). A warning is issued when OT1 encoding is in use at the `\begin{document}`.
- Customisation should now be handled only by command `\frenchbsetup{}`, `frenchb.cfg` (kept for compatibility) should no longer be used. See section 1.2 for the list of available options.
- Captions in figures and tables have changed in French: former abbreviations “Fig.” and “Tab.” have been replaced by full names “Figure” and “Table”. If this leads to formatting problems in captions, you can add the following two commands to your preamble (after loading `babel`) to get the former captions

```
\addto\captionsfrench{\def\figurename{{\scshape Fig.}}}
\addto\captionsfrench{\def\tablename{{\scshape Tab.}}}
```
- The `\nombre` command is now provided by the `numprint` package best loaded 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\bbl@tempa\relax
37 \ifx\magnification\@undefined
38   \ifx\@compatibilitytrue\@undefined
39     \PackageError{frenchb.ldf}
40       {LaTeX-2.09 format is no longer supported.\MessageBreak
41         Aborting here}
42       {Please upgrade to LaTeX2e!}
43   \let\bbl@tempa\endinput
44 \else
45   \LaTeXettrue
46 \fi
47 \fi
48 \bbl@tempa

```

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

```

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

```

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

```

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

```

```
81 \bbl@tempa
```

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

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

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

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

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

\ifFBunicode French hyphenation patterns are now coded in Unicode, see file `hyph-fr.tex`. XeTeX
\ifFBLuaTeX and LuaTeX engines require some extra code to deal with the French “apostrophe”.
\ifFBXeTeX Let's define three new ‘if’: `\ifFBLuaTeX`, `\ifFBXeTeX` 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 T_EX format.

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

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

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

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

```

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

```

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

```

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

```

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

```

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

```

2.2 Punctuation

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

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

```

129 \newif\ifFBAutoSpaceGuill \FBAutoSpaceGuilltrue

```

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

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


```

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

```

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

```

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

```

\FBcolonspace According to the I.N. specifications, the ‘:’ requires an inter-word space before it, the **\FBthinspace** other three require just a `\thinspace`. We define `\FBcolonspace` as `\space` (inter-word space) and `\FBthinspace` as `\thinspace` (both are user customisable). LuaTeX **\FBcolonskip** requires skips instead of commands, so we define `\FBcolonskip` and `\FBthinskip` to hold the specifications (width/stretch/shrink) of `\space` and `\thinspace` for the `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}{\hskip .16667em \relax}
145 \newskip\FBcolonskip
146 \FBcolonskip=3.33pt plus 1.665pt minus 1.11pt \relax
147 \newskip\FBthinskip
148 \FBthinskip=1.66672pt \relax

```

2.2.1 Punctuation with LuaTeX

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

The following `\directlua` call ensures compatibility with LaTeX releases prior to 2015/10/01: the `\lcalleftbox` primitive⁴ introduced by Omega was prefixed with “`luatex`”, it should no longer be, see `ltnews23.tex` for details.

```

149 \ifB@luatex@punct
150   \directlua{tex.enableprimitives("", tex.extraprimitives("omega"))}

```

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

```

151 \begingroup\expandafter\expandafter\expandafter\endgroup
152 \expandafter\ifx\csname newluafunction\endcsname\relax

```

This code is for Plain or LaTeX versions prior to 2015/10/01.

```

153 \ifLaTeXe

```

⁴used by `\frquote`, see p. 33.

```

154     \AtEndOfPackage{%
155         \RequirePackage{luatexbase}%
156         \newluatexattribute\FB@addDPspace \FB@addDPspace=1 \relax
157         \newluatexattribute\FB@addGUIspace \FB@addGUIspace=0 \relax
158     }
159     \else
160         \input luatexbase.sty
161         \newluatexattribute\FB@addDPspace \FB@addDPspace=1 \relax
162         \newluatexattribute\FB@addGUIspace \FB@addGUIspace=0 \relax
163     \fi
164     \else

```

This code is for recent LaTeX versions (starting with 2015/10/01) or Plain when `l\luatex.tex` has been loaded before `babel`.

```

165     \newattribute\FB@addDPspace \FB@addDPspace=1 \relax
166     \newattribute\FB@addGUIspace \FB@addGUIspace=0 \relax
167     \fi
168     \ifLaTeXe
169         \PackageInfo{frenchb.ldf}{No need for active punctuation
170             characters\MessageBreak with this version
171             of LuaTeX!\MessageBreak reported}
172     \else
173         \fb@info{No need for active punctuation characters\\
174             with this version of LuaTeX!}
175     \fi
176 \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).

```

177 local FB_punct_thin =
178   {[string.byte("!")] = true,
179    [string.byte("?")] = true,
180    [string.byte(";")] = true}
181 local FB_punct_thick =
182   {[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 ‘»’.

```

183 local FB_punct_left =
184   {[string.byte("!")] = true,
185    [string.byte("?")] = true,
186    [string.byte(";")] = true,
187    [string.byte(":")] = true,
188    [0x14] = true,
189    [0xBB] = true}
190 local FB_punct_right =

```

```

191  {[0x13]          = true,
192  [0xAB]           = true}

```

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

```

193 local FB_punct_null =
194  {[string.byte("!")] = true,
195   [string.byte("?")] = true,
196   [string.byte("[") ] = true,
197   [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.

```

198  [0xA0]           = true,
199  [0x202F]         = true}
200 local FB_guil_null =
201  {[0xA0]          = true,
202  [0x202F]         = true}

```

Local definitions for nodes:

```

203 local new_node    = node.new
204 local copy_node   = node.copy
205 local node_id     = node.id
206 local HLIST       = node_id("hlist")
207 local TEMP        = node_id("temp")
208 local KERN        = node_id("kern")
209 local GLUE        = node_id("glue")
210 local GSPEC       = node_id("glue_spec")
211 local GLYPH       = node_id("glyph")
212 local PENALTY     = node_id("penalty")
213 local nobreak     = new_node(PENALTY)
214 nobreak.penalty   = 10000
215 local insert_node_before = node.insert_before
216 local insert_node_after  = node.insert_after
217 local remove_node       = node.remove

```

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

```

218 local thin10 = tex.skip['FBthinskip']
219 local thinwd = thin10.width/65536/3.33
220 local thinst = thin10.stretch/65536/1.665
221 local thinsh = thin10.shrink/65536/1.11
222 local coln10 = tex.skip['FBcolonskip']
223 local colnwd = coln10.width/65536/3.33
224 local colnst = coln10.stretch/65536/1.665
225 local colnsh = coln10.shrink/65536/1.11
226 local guil10 = tex.skip['FBguillskip']
227 local guilwd = guil10.width/65536/3.33
228 local guilst = guil10.stretch/65536/1.665
229 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, and of special fonts like `lcircle1.pfb` for which `font.getfont(fid)` does not return a proper font table, in such cases all glue specs will be null):

```

230 local font_table = {}
231 local function new_glue_scaled (fid,width,stretch,shrink)
232   if fid > 0 then
233     local fp = font_table[fid]
234     if not fp then
235       local ft = font.getfont(fid)
236       if ft then
237         font_table[fid] = ft.parameters
238         fp = font_table[fid]
239       end
240     end
241     local gl = new_node(GLUE,0)
242     local gl_spec = new_node(GSPEC)
243     if fp then
244       gl_spec.width = width * fp.space
245       gl_spec.stretch = stretch * fp.space_stretch
246       gl_spec.shrink = shrink * fp.space_shrink
247     end
248     gl.spec = gl_spec
249     return gl
250   else
251     return nil
252   end
253 end

```

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

```

254 local addDPspace   = luatexbase.attributes['FB@addDPspace']
255 local addGUILspace = luatexbase.attributes['FB@addGUILspace']
256 local has_attribute = node.has_attribute

```

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

```

257 local function french_punctuation (head)
258   for item in node.traverse_id(GLYPH, head) do
259     local lang = item.lang
260     local char = item.char
261     local fid  = item.font
262     local SIG  = has_attribute(item, addGUILspace)
263     if lang == FR and FB_punct_left[char] and fid > 0 then
264       local prev = item.prev
265       local prev_id, prev_subtype, prev_char
266       if prev then

```

```

267         prev_id = prev.id
268         prev_subtype = prev.subtype
269         if prev_id == GLYPH then
270             prev_char = prev.char
271         end
272     end

```

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

```

273     local glue = prev_id == GLUE and prev_subtype == 0
274     local glue_wd
275     if glue then
276         glue_spec = prev.spec
277         glue_wd = glue_spec.width
278     end
279     local realglue = glue and glue_wd > 1

```

For characters for which `FB_punct_thin` or `FB_punct_thick` is *true*, the amount of spacing to be typeset before them is controlled by `\FBthinskip` (`thinwd`, `thinst`, `thinsh`) or `\FBcolonskip` (`colnwd`, `colnst`, `colnsh`) respectively. Two options: if a space has been typed in before (turned to *glue* in the node list), we remove the *glue* and add a nobreak penalty and the required *glue*. Otherwise (auto option), the penalty and the required *glue* are inserted if attribute `\FB@addDPspace` is set, unless one of these three condition is met: a) the previous character is part of type `FB_punct_null` (this avoids spurious spaces in strings like (!) or ??), b) a null glue (actually glues <= 1 sp for tabulars) preceeds the punctuation character, c) the punctuation character starts a paragraph or an `\hbox{}`.

```

280     if FB_punct_thin[char] or FB_punct_thick[char] then
281         local SBDP = has_attribute(item, addDPspace)
282         local auto = SBDP and SBDP > 0
283         if auto then
284             if (prev_char and FB_punct_null[prev_char]) or
285                 (glue and glue_wd <= 1) or
286                 (prev_id == HLIST and prev_subtype == 3) or
287                 (prev_id == TEMP) then
288                 auto = false
289             end
290         end
291         local fbglue
292         if FB_punct_thick[char] then
293             fbglue = new_glue_scaled(fid,colnwd,colnst,colnsh)
294         else
295             fbglue = new_glue_scaled(fid,thinwd,thinst,thinsh)
296         end
297         if realglue or auto then
298             if realglue then
299                 head = remove_node(head,prev,true)
300             end
301             insert_node_before(head, item, copy_node(nobreak))
302             insert_node_before(head, item, copy_node(fbglue))
303         end

```

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

```

304     elseif SIG and SIG > 0 then
305         local addgl = (prev_char and not FB_guil_null[prev_char]) or
306                       (not prev_char and
307                        prev_id ~= TEMP and
308                        not (prev_id == HLIST and prev_subtype == 3)
309                       )

```

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

```

310         if glue and glue_wd <= 1 then
311             addgl = false
312         end
313         if addgl then
314             if glue then
315                 head = remove_node(head,prev,true)
316             end
317             local fbglue = new_glue_scaled(fid,guilwd,guilst,guilsh)
318             insert_node_before(head, item, copy_node(nobreak))
319             insert_node_before(head, item, copy_node(fbglue))
320         end
321     end
322 end

```

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

```

323     if lang == FR and FB_punct_right[char] and fid > 0
324                                     and SIG and SIG > 0 then
325         local next = item.next
326         local next_id, next_subtype, next_char, nextnext, kern_wd
327         if next then
328             next_id = next.id
329             next_subtype = next.subtype
330             if next_id == GLYPH then
331                 next_char = next.char

```

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

```

332     elseif next_id == KERN then
333         kern_wd = next.kern
334         if kern_wd == 0 then
335             nextnext = next.next

```

```

336         if nextnext then
337             next = nextnext
338             next_id = nextnext.id
339             next_subtype = nextnext.subtype
340             if next_id == GLYPH then
341                 next_char = nextnext.char
342             end
343         end
344     end
345 end
346
347 local glue = next_id == GLUE and next_subtype == 0
348 if glue then
349     glue_spec = next.spec
350     glue_wd = glue_spec.width
351 end
352 local addgl = (next_char and not FB_guil_null[next_char]) or
353              (next and not next_char)

```

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

```

354     if glue and glue_wd == 0 then
355         addgl = false
356     end
357     if addgl then
358         if glue then
359             head = remove_node(head,next,true)
360         end
361         local fid = item.font
362         local fbglue = new_glue_scaled(fid,guilwd,guilst,guilsh)
363         insert_node_after(head, item, copy_node(fbglue))
364         insert_node_after(head, item, copy_node(nobreak))
365     end
366 end
367 end
368 return head
369 end
370 return french_punctuation

```

`\FB@luatex@punct@french` 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.

```

371 \ifFB@luatex@punct
372   \newcommand*{\FB@luatex@punct@french}{%
373     \babel@save{\shorthandon}%
374     \babel@save{\shorthandoff}%
375     \def\shorthandoff##1{%
376       \ifx\PackageWarning\undefined
377         \fb@warning{\noexpand\shorthandoff{;:!?}} is helpless with
378         LuaTeX,\\ use \noexpand\NoAutoSpacing

```

```

379         *inside a group* instead.}%
380     \else
381         \PackageWarning{frenchb.ldf}{\protect\shorthandoff{;:!?} is
382         helpless with LuaTeX,\MessageBreak use \protect\NoAutoSpacing
383         \space *inside a group* instead;\MessageBreak reported}%
384     \fi}%
385     \def\shorthandon##1{%
386 }
387 \FB@addto{extras}{\FB@luatex@punct@french}

```

In \LaTeX 2_ϵ , file `frenchb.lua` will be loaded ‘AtBeginDocument’ *after* processing options (`ThinColonSpace` needs to be taken into account). The next definition will be used to activate Lua punctuation: it sets the language number for French, loads `frenchb.lua` and adds function `french_punctuation` at the end of the kerning callback (no priority).

```

388 \def\activate@luatexpunct{%
389     \directlua{%
390         FR = \the\l@french
391         local path = kpse.find_file("frenchb.lua", "lua")
392         if path then
393             local f = dofile(path)
394             luatexbase.add_to_callback("kerning",
395                                     f, "frenchb.french_punctuation")
396         else
397             texio.write_nl('')
398             texio.write_nl('*****')
399             texio.write_nl('Error: frenchb.lua not found.')
400             texio.write_nl('*****')
401             texio.write_nl('')
402         end
403     }%
404 }
405 \fi

```

End of specific code for punctuation with LuaTeX engines.

2.2.2 Punctuation with XeTeX

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

The default value for `\XeTeXcharclass` is 0 for characters tokens and 255 for all other tokens (glues, kerns, math and box boundaries, etc.). These defaults should not be changed 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.

```

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

```

Six new character classes are defined for frenchb.

```

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

```

As `\babel@savevariable` doesn't work inside a `\bbl@for` loop, we define a variant to save the `\XeTeXcharclass` values which will be modified in French.

```

421 \def\FBsavevariable@loop#1#2{\begingroup
422   \toks@\expandafter{\originalTeX #1}%
423   \edef\x{\endgroup
424     \def\noexpand\originalTeX{\the\toks@ #2=\the#1#2\relax}}%
425   \x}

```

`\FB@charlist` holds the all list of characters which have their `\XeTeXcharclass` value modified in French: the first set includes high punctuation, French quotes, opening delimiters and no-break spaces

"21	"3A	"3B	"3F	"AB	"BB	"28	"5B	"A0	"202F
!	:	;	?	«	»	([

the second one holds those which need resetting in French when `xeCJK.sty` is in use

"29	"5D	"7B	"7D	"2C	"2D	"2E	"22	"25	"27	"60	"2019
)]	{	}	,	-	.	"	%	'	'	'

```

426 \def\FB@charlist{"21,"3A,"3B,"3F,"AB,"BB,"28,"5B,"A0,"202F,%
427   "29,"5D,"7B,"7D,"2C,"2D,"2E,"22,"25,"27,"60,"2019}

```

`\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.

```

428 \newcommand*{\FB@xetex@punct@french}{%
429   \babel@savevariable{\XeTeXinterchartokenstate}%
430   \babel@save{\shorthandon}%
431   \babel@save{\shorthandoff}%
432   \bbl@for\FB@char\FB@charlist
433     {\FBsavevariable@loop{\XeTeXcharclass}{\FB@char}}%
434   \def\shorthandoff##1{%

```

```

435 \ifx\PackageWarning\@undefined
436 \fb@warning{\noexpand\shorthandoff{;:!?} is helpless with
437 XeTeX,\ use \noexpand\NoAutoSpacing
438 *inside a group* instead.}%
439 \else
440 \PackageWarning{frenchb.ldf}{\protect\shorthandoff{;:!?} is
441 helpless with XeTeX,\MessageBreak use \protect\NoAutoSpacing
442 \space *inside a group* instead;\MessageBreak reported}%
443 \fi}%
444 \def\shorthandon##1{}%

```

Let's now set the classes and interactions between classes.

```

445 \XeTeXinterchartokenstate=1
446 \XeTeXcharclass '\: = \FB@punctthick
447 \XeTeXinterchartoks \z@ \FB@punctthick = {%
448 \ifhmode\FDP@colonspace\fi}%
449 \XeTeXinterchartoks \FB@guilf \FB@punctthick = {%
450 \FDP@colonspace}%

```

Small glues such as “glue 1sp” in tabular ‘l’ columns or “glue 0 plus 1 fil” in tabular ‘c’ columns or lstlisting environment should not trigger any extra space; they will still do when [AutoSpacePunctuation](#) is true: unfortunately \XeTeXcharclass=255 isn't specific to glue tokens (this class includes box and math boundaries f.i.), so the \else part cannot be omitted.

```

451 \XeTeXinterchartoks 255 \FB@punctthick = {%
452 \ifhmode
453 \ifdim\lastskip>1sp
454 \unskip\penalty\@M\FBcolonspace
455 \else
456 \FDP@colonspace
457 \fi
458 \fi}%
459 \bbl@for\FB@char
460 {\';, '\!, '\?}%
461 {\XeTeXcharclass\FB@char=\FB@punctthin}%
462 \XeTeXinterchartoks \z@ \FB@punctthin = {%
463 \ifhmode\FDP@thinspace\fi}%
464 \XeTeXinterchartoks \FB@guilf \FB@punctthin = {%
465 \FDP@thinspace}%
466 \XeTeXinterchartoks 255 \FB@punctthin = {%
467 \ifhmode
468 \ifdim\lastskip>1sp
469 \unskip\penalty\@M\FBthinspace
470 \else
471 \FDP@thinspace
472 \fi
473 \fi}%
474 \XeTeXinterchartoks \FB@guilo \z@ = {%
475 \ifFBAutoSpaceGuill\FBguillspace\fi}%
476 \XeTeXinterchartoks \FB@guilo 255 = {%
477 \ifFBAutoSpaceGuill\FBguillspace\ignorespaces\fi}%

```

```

478 \XeTeXinterchartoks \z@ \FB@guilf = {%
479     \ifFBAutoSpaceGuill\FBguillspace\fi}%
480 \XeTeXinterchartoks \FB@punctthin \FB@guilf = {%
481     \ifFBAutoSpaceGuill\FBguillspace\fi}%
482 \XeTeXinterchartoks 255 \FB@guilf = {%
483     \ifFBAutoSpaceGuill\unskip\FBguillspace\fi}%

```

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

```

484 \bbl@for\FB@char
485     {\[, '\[, "A0, "202F}%
486     {\XeTeXcharclass\FB@char=\FB@punctnul}%

```

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

```

487 \bbl@for\FB@char
488     {\[, '\[, '\., '\-, '\), '\], '\}, '\%, "22, "27, "60, "2019}%
489     {\XeTeXcharclass\FB@char=\z@}%
490 }
491 \FB@addto{extras}{\FB@xetex@punct@french}

```

End of specific code for punctuation with modern XeTeX engines.

```
492 \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.

```

493 \ifFB@active@punct
494 \initiate@active@char{:}%
495 \initiate@active@char{;}%
496 \initiate@active@char{!}%
497 \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`.

```

498 \declare@shorthand{french}{;}{;%
499     \ifhmode
500         \ifdim\lastskip>1sp
501             \unskip\penalty\@M\FBthinspace
502         \else
503             \FDP@thinspace
504         \fi
505     \fi

```

Now we can insert a ; character.

```
506 \string;}

```

The next three definitions are very similar.

```

507 \declare@shorthand{french}{!}{%
508     \ifhmode
509         \ifdim\lastskip>1sp
510             \unskip\penalty\@M\FBthinspace
511         \else
512             \FDP@thinspace
513         \fi
514     \fi
515     \string!}
516 \declare@shorthand{french}{?}{%
517     \ifhmode
518         \ifdim\lastskip>1sp
519             \unskip\penalty\@M\FBthinspace
520         \else
521             \FDP@thinspace
522         \fi
523     \fi
524     \string?}
525 \declare@shorthand{french}{:}{%
526     \ifhmode
527         \ifdim\lastskip>1sp
528             \unskip\penalty\@M\FBcolonspace
529         \else
530             \FDP@colonspace
531         \fi
532     \fi
533     \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.

```

534 \declare@shorthand{system}{:}{\string:}
535 \declare@shorthand{system}{!}{\string!}
536 \declare@shorthand{system}{?}{\string?}
537 \declare@shorthand{system}{;}{\string;}
538 %}

```

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

```

539 \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.

```

540     \bbl@activate{:}\bbl@activate{;}%
541     \bbl@activate{!}\bbl@activate{?}%
542 }
543 \FB@addto{noextras}{%
544     \bbl@deactivate{:}\bbl@deactivate{;}%
545     \bbl@deactivate{!}\bbl@deactivate{?}%
546 }

```

547 \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.

548 \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).

```
549 \def\autospace@beforeFDP{%
550     \ifFB@luatex@punct\FB@addDPSpace=1 \fi
551     \def\FDP@thinspace{\penalty\@M\FBthinspace}%
552     \def\FDP@colonspace{\penalty\@M\FBcolonspace}}
553 \def\noautospace@beforeFDP{%
554     \ifFB@luatex@punct\FB@addDPSpace=0 \fi
555     \let\FDP@thinspace\@empty
556     \let\FDP@colonspace\@empty}
557 \ifLaTeXe
558     \def\AutoSpaceBeforeFDP{\autospace@beforeFDP
559                             \FBAutoSpacePunctuationtrue}
560     \def\NoAutoSpaceBeforeFDP{\noautospace@beforeFDP
561                             \FBAutoSpacePunctuationfalse}
562     \AtEndOfPackage{\AutoSpaceBeforeFDP}
563 \else
564     \let\AutoSpaceBeforeFDP\autospace@beforeFDP
565     \let\NoAutoSpaceBeforeFDP\noautospace@beforeFDP
566     \AutoSpaceBeforeFDP
567 \fi
```

\rmfamilyFB In L^AT_EX 2_ε \ttfamily (and hence \texttt) will be redefined ‘AtBeginDocument’ as **\sffamilyFB** \ttfamilyFB so that no space is added before the four ; : ! ? characters, even **\ttfamilyFB** 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.

568 \ifLaTeXe

```

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

```

592 \newcommand*{\NoAutoSpacing}{\FBAutoSpaceGuillfalse
593   \ifFB@active@punct\shorthandoff{;:!?}\fi
594   \ifFB@xetex@punct\XeTeXinterchartokenstate=0 \fi
595   \ifFB@luatex@punct\FB@addDPspace=0 \FB@addGUILspace=0 \fi
596 }

```

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**.

```

597 \newcommand*{\og}{\@empty}
598 \newcommand*{\fg}{\@empty}

```

\guillemotleft 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.

```

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

```

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

```

`\FBguillspace` is not used with LuaTeX.

```

621 \ifFB@luatex@punct
622   \DeclareRobustCommand*{\FB@og}{\leavevmode
623     \bgroup\FB@addGUllspace=1 \guillemotleft\egroup}
624   \DeclareRobustCommand*{\FB@fg}{\ifdim\lastskip>\z@\unskip\fi
625     \bgroup\FB@addGUllspace=1 \guillemotright\egroup\xspace}
626 \fi

```

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

```

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

```

643 \ifLaTeXe
644   \def\bbf@frenchguillemets{\renewcommand*{\og}{\FB@og}%
645     \renewcommand*{\fg}{\FB@fg}}
646   \renewcommand*{\og}{\textquotedblleft}
647   \renewcommand*{\fg}{\ifdim\lastskip>z@\unskip\fi \textquotedblright}
648 \else
649   \def\bbf@frenchguillemets{\let\og\FB@og
650     \let\fg\FB@fg}
651   \def\og{\textquotedblleft}
652   \def\fg{\ifdim\lastskip>z@\unskip\fi\textquotedblright}
653 \fi
654 \FB@addto{extras}{\babel@save\og \babel@save\fg \bbf@frenchguillemets}

```

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

```

655 \newcommand*{\ogi}{\FB@og}
656 \newcommand*{\fgi}{\FB@fg}
657 \newcommand*{\ogii}{\textquotedblleft}
658 \newcommand*{\fgii}{\textquotedblright}

```

and the needed technical stuff to handle options:

```

659 \newcount\FBguill@level
660 \newtoks\FB@everypar
661 \newif\ifFBcloseguill \FBcloseguilltrue
662 \newif\ifFBInnerGuillSingle
663 \def\FBguillopen{\bgroup\NoAutoSpacing\guillemotleft\egroup}
664 \def\FBguillclose{\bgroup\NoAutoSpacing\guillemotright\egroup}
665 \let\FBguillnone\relax
666 \let\FBeveryparguill\FBguillopen
667 \ifFB@luatex@punct
668   \let\FBverylineguill\FBguillopen

```



```

669 \else
670   \let\FBverylineguill\FBguillnone
671 \fi

```

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

```

672 \ifLaTeXe
673   \DeclareRobustCommand\frquote{%
674     \ifstar{\FBcloseguillfalse\fr@quote}%
675     {\FBcloseguilltrue\fr@quote}}
676 \else
677   \newcommand\frquote[1]{\fr@quote{#1}}
678 \fi

```

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

```

679 \newcommand{\fr@quote}[1]{%
680   \leavevmode
681   \advance\FBguill@level by \@ne

```

Kern used inside French quotes; must match the fixed part of `\FBguillspace`.

```

682   \def\FB@quotespace{\kern.8\fontdimen2\font}%
683   \ifcase\FBguill@level
684   \or

```

This for level 1 (outer) quotations: save `\everypar` before customising it, set `\FBeverypar@quote` for level 1 quotations and add it to `\everypar`, then print the quotation:

```

685   \FB@everypar=\everypar
686   \ifx\FBveryparguill\relax
687   \else
688     \def\FBverypar@quote{\FBveryparguill\FB@quotespace}%
689     \everypar=\expandafter{\the\everypar \FBverypar@quote}%
690   \fi
691   \logi #1\fgi
692 \or

```

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

```

693   \ifx\FBverylineguill\FBguillopen
694     \localleftbox{\guillemotleft\FB@quotespace}%
695     \let\FBverypar@quote\relax
696     \logi #1\ifFBcloseguill\fgi\fi
697   \else
698     \ifx\FBverylineguill\FBguillclose
699       \localleftbox{\guillemotright\FB@quotespace}%
700       \let\FBverypar@quote\relax
701       \logi #1\ifFBcloseguill\fgi\fi
702   \else

```

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

```

703      \let\FBeverypar@quote\relax
704      \ifFBInnerGuillSingle
705          \def\ogii{\leavevmode
706              \guilsinglleft\FBguillspace}%
707          \def\fgii{\ifdim\lastskip>\z@\unskip\fi
708              \FBguillspace\guilsinglright}%
709          \ifx\FBeveryparguill\FBguillopen
710              \def\FBeverypar@quote{\guilsinglleft\FB@quotespace}%
711          \fi
712          \ifx\FBeveryparguill\FBguillclose
713              \def\FBeverypar@quote{\guilsinglright\FB@quotespace}%
714          \fi
715      \fi
716      \ogii #1\ifFBcloseguill \fgii \fi
717  \fi
718  \fi
719  \else

```

Warn if `\FBguill@level` ≥ 3 :

```

720      \ifx\PackageWarning\@undefined
721          \fb@warning{\noexpand\frquote\space accepts no more than
722              two levels.\ Quotation not printed.}%
723      \else
724          \PackageWarning{frenchb.ldb}{%
725              \protect\frquote\space accepts no more than two levels
726              \MessageBreak Quotation not printed. Reported}
727      \fi
728  \fi

```

Clean on exit: adjust `\FBguill@level` and restore `\localleftbox` and `\everypar`.

```

729  \advance\FBguill@level by \m@ne
730  \ifx\FBeverylineguill\FBguillnone\else\localleftbox{}\fi
731  \ifx\FBeveryparguill\relax\else\everypar=\FB@everypar\fi
732 }

```

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.

```

733 \ifLaTeXe
734   \def\BabelLanguages{french,acadian}
735   \StartBabelCommands*\BabelLanguages\{date}
736   [unicode, fontenc=EU1 EU2, charset=utf8]
737   \SetString\monthiiname{février}
738   \SetString\monthviiiname{août}
739   \SetString\monthxiiname{décembre}

```

```

740 \StartBabelCommands*{\BabelLanguages}{date}
741 \SetStringLoop{month#lname}{%
742     janvier,f\'evrier,mars,avril,mai,juin,juillet,%
743     ao\^ut,septembre,octobre,novembre,d\'ecembre}
744 \SetString\today{{\number\day}\ifnum1=\day {\ier}\fi\space
745     \csname month\romannumeral\month name\endcsname \space
746     \number\year
747 }
748 \EndBabelCommands
749 \else
750 \ifFBunicode
751 \namedef{date\CurrentOption}{%
752     \def\today{{\number\day}\ifnum1=\day {\ier}\fi \space
753     \ifcase\month
754         \or janvier\or février\or mars\or avril\or mai\or
755         juin\or juillet\or août\or septembre\or
756         octobre\or novembre\or décembre\fi
757     \space \number\year}}
758 \else
759 \namedef{date\CurrentOption}{%
760     \def\today{{\number\day}\ifnum1=\day {\ier}\fi \space
761     \ifcase\month
762         \or janvier\or f\'evrier\or mars\or avril\or mai\or
763         juin\or juillet\or ao\^ut\or septembre\or
764         octobre\or novembre\or d\'ecembre\fi
765     \space \number\year}}
766 \fi
767 \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 \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).

```

768 \newif\ifFB@poorman
769 \newdimen\FB@Mht
770 \ifLaTeXe
771 \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{}`.

```

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

```

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.

```

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

```

788 \edef\reserved@a{\lowercase{%
789   \noexpand\IfFileExists{\f@encoding\FB@firstthree -sup.fd}}}%
790 \reserved@a
791 {\ifx\FB@suffix\FB@x \FB@poormanfalse\fi
792  \ifx\FB@suffix\FB@j \FB@poormanfalse\fi
793  \if\FB@poorman \FB@up@fake{#1}%
794  \else          \FB@up@real{#1}%
795  \fi}%
796 {\FB@up@fake{#1}}%

```

```

797 \egroup}
\FB@up@real just picks up the superscripts from the subfamily (and forces lower-
case).
798 \newcommand*{\FB@up@real}[1]{\bgroup
799 \fontfamily{\FB@firstthree -sup}\selectfont \FB@lc{#1}\egroup}
\fbup is defined as \FB@up unless \realsuperscript is defined by realscripts.sty.
800 \DeclareRobustCommand*{\fbup}[1]{%
801 \ifx\realsuperscript\@undefined
802 \FB@up{#1}%
803 \else
804 \bgroup\let\fakesuperscript\FB@up@fake
805 \realsuperscript{\FB@lc{#1}}\egroup
806 \fi}
Let's provide a temporary definition for \up (redefined 'AtBeginDocument' as \fbup or
\textsuperscript according to \frenchbsetup{} options).
807 \providecommand*{\up}{\relax}
Poor man's definition of \up for Plain.
808 \else
809 \providecommand*{\up}[1]{\leavevmode\raise1ex\hbox{\sevenrm #1}}
810 \fi

```

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

```

\ier 811 \def\ieme{\up{e}\xspace}
\iere 812 \def\iemes{\up{es}\xspace}
\iemes 813 \def\ier{\up{er}\xspace}
\iers 814 \def\iers{\up{ers}\xspace}
\ieres 815 \def\iere{\up{re}\xspace}
816 \def\ieres{\up{res}\xspace}

```

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

```

\no 817 \newcommand*{\FrenchEnumerate}[1]{%
\nos 818 #1\up{o}\kern+.3em}
\nos 819 \newcommand*{\FrenchPopularEnumerate}[1]{%
\primo 820 #1\up{o})\kern+.3em}
\frprimo) Typing \primo should result in '1° ',
821 \def\primo{\FrenchEnumerate1}
822 \def\secundo{\FrenchEnumerate2}
823 \def\tertio{\FrenchEnumerate3}
824 \def\quarto{\FrenchEnumerate4}
while typing \frprimo) gives '1°) .
825 \def\frprimo){\FrenchPopularEnumerate1}
826 \def\frsecundo){\FrenchPopularEnumerate2}
827 \def\frtertio){\FrenchPopularEnumerate3}
828 \def\frquarto){\FrenchPopularEnumerate4}

```

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

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

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

```
836 \ifFBunicode
837   \newcommand*\at{{\char"0040}}
838   \newcommand*\circonflexe{{\char"005E}}
839   \newcommand*\tild{{\char"007E}}
840   \newcommand*\boi{\textbackslash}
841   \newcommand*\degre{{\char"00B0}}
842 \else
843   \ifLaTeXe
844     \DeclareTextSymbol{\at}{T1}{64}
845     \DeclareTextSymbol{\circonflexe}{T1}{94}
846     \DeclareTextSymbol{\tild}{T1}{126}
847     \DeclareTextSymbolDefault{\at}{T1}
848     \DeclareTextSymbolDefault{\circonflexe}{T1}
849     \DeclareTextSymbolDefault{\tild}{T1}
850     \DeclareRobustCommand*\boi{\textbackslash}
851     \DeclareRobustCommand*\degre{\r{}}
852   \else
853     \def\T@one{T1}
854     \ifx\fontencoding\T@one
855       \newcommand*\degre{{\char6}}
856     \else
857       \newcommand*\degre{{\char23}}
858     \fi
859     \newcommand*\at{{\char64}}
860     \newcommand*\circonflexe{{\char94}}
861     \newcommand*\tild{{\char126}}
862     \newcommand*\boi{\backslash}
863   \fi
864 \fi
```

\degrees We now define a macro `\degrees` for typesetting the abbreviation for ‘degrees’ (as in ‘degrees Celsius’). As the bounding box of the character ‘degree’ has very different widths in CM/EC and PostScript fonts, we fix the width of the bounding box of `\degrees` to 0.3 em, this lets the symbol ‘degree’ stick to the preceding (e.g., 45\degrees) or following character (e.g., 20~\degrees C).

If T_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.

```

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

```

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 `{,}`. **\StandardMathComma** 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.

```

891 \newcount\std@mcc
892 \newcount\dec@mcc
893 \std@mcc=\mathcode'\,
894 \dec@mcc=\std@mcc
895 \@tempcnta=\std@mcc
896 \divide\@tempcnta by "1000

```

```

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

```

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

```

907 \newcommand*\nombre[1]{\iflanguage{french}%
908     {\fb@warning{*** \noexpand\nombre
no longer formats numbers\string! ***}}

```

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

```

909 \let\FBstop@here\relax
910 \def\FBclean@on@exit{\let\ifLaTeXe\undefined
911     \let\LaTeXettrue\undefined
912     \let\LaTeXefalse\undefined}
913 \ifx\magnification\@undefined
914 \else
915     \def\FBstop@here{\ifFB@luatex@punct
916         \activate@luatexpunct
917     \fi
918     \FBclean@on@exit
919     \ldf@quit\CurrentOption\endinput}
920 \fi
921 \FBstop@here

```

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

```

922 \renewcommand*\nombre[1]{\Warning@nombre{#1}}
923 \newcommand*\Warning@nombre[1]{%
924     \ifdefined\numprint
925         \numprint{#1}%
926     \else
927         \PackageWarning{frenchb.ldf}{%
928             \protect\nombre\space now relies on package numprint.sty,%
929             \MessageBreak add \protect
930             \usepackage[autolanguage]{numprint},\MessageBreak

```



```

931         see file numprint.pdf for more options.\MessageBreak
932         \protect\nombre\space called}%
933     \global\let\Warning@nombre\relax
934     {#1}%
935 \fi
936 }

```

2.7 Caption names

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

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

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

```

937 \ifx\FBfigtabshape\undefined \let\FBfigtabshape\scshape \fi

```

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

```

938 \StartBabelCommands*{\BabelLanguages}{captions}
939     [unicode, fontenc=EU1 EU2, charset=utf8]
940     \SetString{\refname}{Références}
941     \SetString{\abstractname}{Résumé}
942     \SetString{\prefacename}{Préface}
943     \SetString{\contentsname}{Table des matières}
944     \SetString{\ccname}{Copie à }
945     \SetString{\proofname}{Démonstration}
946     \SetStringLoop{ordinal#1}{%
947         Première,Deuxième,Troisième,Quatrième,Cinquième,%
948         Sixième,Septième,Huitième,Neuvième,Dixième,Onzième,%
949         Douzième,Treizième,Quatorzième,Quinzième,Seizième,%
950         Dix-septième,Dix-huitième,Dix-neuvième,Vingtième}
951 \StartBabelCommands*{\BabelLanguages}{captions}
952     \SetString{\refname}{R\`ef\`erences}
953     \SetString{\abstractname}{R\`esum\`e}
954     \SetString{\bibname}{Bibliographie}
955     \SetString{\prefacename}{Pr\`eface}
956     \SetString{\chaptername}{Chapitre}
957     \SetString{\appendixname}{Annexe}
958     \SetString{\contentsname}{Table des mati\`eres}
959     \SetString{\listfigurename}{Table des figures}
960     \SetString{\listtablename}{Liste des tableaux}
961     \SetString{\indexname}{Index}
962     \SetString{\figurename}{{\FBfigtabshape Figure}}
963     \SetString{\tablename}{{\FBfigtabshape Table}}
964     \SetString{\pagename}{page}
965     \SetString{\seename}{voir}
966     \SetString{\alsoname}{voir aussi}
967     \SetString{\enclname}{P.~J. }
968     \SetString{\ccname}{Copie \`a }

```

```

969 \SetString{\headtoname}{}
970 \SetString{\proofname}{Démonstration}
971 \SetString{\glossaryname}{Glossaire}

```

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

```

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

```

The following patch is for koma-script classes: `\partformat` needs to be redefined in French as this command, defined as `\partname~\thepart\autodot` is incompatible with our redefinition of `\partname`. The code is postponed to the end of package because `\ifFB@koma` will be defined and set later on (see p. 44).

```

989 \AtEndOfPackage{%
990     \ifFB@koma
991         \ifdefined\partformat
992             \FB@addto{captions}{%
993                 \ifFBPartNameFull
994                     \babel@save\partformat
995                     \renewcommand*\partformat{\partname}%
996                 \fi}%
997         \fi
998     \fi
999 }

```

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

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

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

```

1000 \bgroup
1001 \catcode'::=12 \catcode'>:=12 \relax
1002 \long\gdef\STD@makecaption#1#2{%
1003   \vskip\abovcaptionskip
1004   \sbox\@tempboxa{#1: #2}%
1005   \ifdim \wd\@tempboxa >\hsize
1006     #1: #2\par
1007   \else
1008     \global \@minipagefalse
1009     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
1010   \fi
1011   \vskip\belowcaptionskip}
1012 \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.

```

1013 \newif\if@FBwarning@capsep
1014 \@FBwarning@capseptrue
1015 \newcommand{\FBWarning}[2]{\PackageWarning{#1}{#2}}
1016 \newcommand*{\CaptionSeparator}{\space\textendash\space}
1017 \def\FBCaption@Separator{: }
1018 \long\def\FB@makecaption#1#2{%
1019   \vskip\abovcaptionskip
1020   \sbox\@tempboxa{#1\FBCaption@Separator #2}%
1021   \ifdim \wd\@tempboxa >\hsize
1022     #1\FBCaption@Separator #2\par
1023   \else
1024     \global \@minipagefalse
1025     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
1026   \fi
1027   \vskip\belowcaptionskip}

```

Disable the standard warning with AMS and SMF classes.

```
1028 \ifclassloaded{amsart}{\@FBwarning@capsepfalse}{}
1029 \ifclassloaded{amsbook}{\@FBwarning@capsepfalse}{}
1030 \ifclassloaded{amsdtx}{\@FBwarning@capsepfalse}{}
1031 \ifclassloaded{amslatex}{\@FBwarning@capsepfalse}{}
1032 \ifclassloaded{amproc}{\@FBwarning@capsepfalse}{}
1033 \ifclassloaded{smfart}{\@FBwarning@capsepfalse}{}
1034 \ifclassloaded{smfbook}{\@FBwarning@capsepfalse}{}

```

Disable the standard warning unless high punctuation is active.

```
1035 \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).

```
1036 \newif\ifFB@koma
1037 \ifclassloaded{memoir}{\@FBwarning@capsepfalse}{}
1038 \ifclassloaded{scrartcl}{\@FBwarning@capsepfalse\FB@komatrue}{}
1039 \ifclassloaded{scrbook}{\@FBwarning@capsepfalse\FB@komatrue}{}
1040 \ifclassloaded{scrreprt}{\@FBwarning@capsepfalse\FB@komatrue}{}

```

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

```
1041 \ifclassloaded{beamer}{\@FBwarning@capsepfalse}{}
1042 \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.

```
1043 \ifpackageloaded{caption}
1044   {\FBWarning{frenchb.lda}%
1045     {Please load the "caption" package\MessageBreak
1046       AFTER babel/frenchb; reported}%
1047   \@FBwarning@capsepfalse}%
1048 {}

```

Same for package floatrow.

```
1049 \ifpackageloaded{floatrow}
1050   {\FBWarning{frenchb.lda}%
1051     {Please load the "floatrow" package\MessageBreak
1052       AFTER babel/frenchb; reported}%
1053   \@FBwarning@capsepfalse}%
1054 {}

```

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').

```
1055 \AtBeginDocument{%
1056   \ifx\@makecaption\STD@makecaption
1057     \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.

```

1058     \ifFBoldFigTabCaptions
1059     \else
1060         \def\FBCaption@Separator{{\autospace@beforeFDP : }}%
1061     \fi
1062     \ifFBCustomiseFigTabCaptions
1063         \ifx\bbl@main@language\FB@french
1064             \def\FBCaption@Separator{\CaptionSeparator}%
1065         \fi
1066     \fi
1067     \@FBwarning@capsepfalse
1068 \fi
1069 \if@FBwarning@capsep
1070     \FBWarning{frenchb.ldb}%
1071     {Figures' and tables' captions might look like\MessageBreak
1072     'Figure 1:' which is wrong in French.\MessageBreak
1073     Check your class or packages to change this;\MessageBreak
1074     reported}%
1075 \fi
1076 \let\FB@makecaption\relax
1077 \let\STD@makecaption\relax
1078 }

```

2.8 Dots...

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

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

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

```

1079 \ifFBunicode
1080     \let\FBtextellipsis\textellipsis
1081 \else
1082     \DeclareTextSymbol{\FBtextellipsis}{LY1}{133}
1083     \DeclareTextCommandDefault{\FBtextellipsis}{%
1084         .\kern\fontdimen3\font.\kern\fontdimen3\font.\xspace}
1085 \fi

```

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

```

1086 \newcommand*{\Tdots@}{\@xp\textellipsis}
1087 \newcommand*{\Mdots@}{\@xp\mdots@}
1088 \AtBeginDocument{\DeclareRobustCommand*{\dots}{\relax
1089             \csname\ifmode M\else T\fi dots@\endcsname}%
1090             \ifdefined\@xp\else\let\@xp\relax\fi
1091             \ifdefined\mdots@\else\let\Mdots@\mathellipsis\fi
1092             }
1093 \def\bbl@frenchdots{\babel@save\Tdots@ \let\Tdots@\FBtextellipsis}
1094 \FB@addto{extras}{\bbl@frenchdots}

```

2.9 More checks about packages' loading order

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

```

1095 \ifFB@active@punct
1096   \@ifpackageloaded{listings}
1097     {\FBWarning{frenchb.ldf}%
1098       {Please load the "listings" package\MessageBreak
1099         AFTER babel/frenchb; reported}%
1100     }{}
1101 \fi

```

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

```

1102 \newif\if@FBwarning@natbib
1103 \ifFB@active@punct
1104   \@ifpackageloaded{natbib}{\@FBwarning@natbibtrue}
1105 \fi
1106 \AtBeginDocument{%
1107   \if@FBwarning@natbib
1108     \@ifpackageloaded{natbib}{\@FBwarning@natbibfalse}%
1109   \fi
1110   \if@FBwarning@natbib
1111     \FBWarning{frenchb.ldf}%
1112     {Please load the "natbib" package\MessageBreak
1113       BEFORE babel/frenchb; reported}%
1114   \fi
1115 }

```

2.10 Setup options: keyval stuff

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

Option processing can occur either in `\frenchbsetup{}`, but *only for options explicitly set* by `\frenchbsetup{}`, or 'AtBeginDocument'; any option affecting `\extrasfrench{}` *must* be processed by `\frenchbsetup{}`: when French is the main language, `\extrasfrench{}` is executed by babel when it switches the main

language and this occurs *before* reading the stuff postponed by frenchb ‘AtBegin-Document’. Reexecuting `\extrasfrench{}` is an option 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` did not work for French).

`\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.

```
1116 \newcommand*{\frenchbsetup}[1]{%
1117   \setkeys{FB}{#1}%
1118 }%
1119 \@onlypreamble\frenchbsetup
```

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

```
1120 \newif\ifFBShowOptions           \FBShowOptionsfalse
1121 \newif\ifFBStandardLayout        \FBStandardLayouttrue
1122 \newif\ifFBGlobalLayoutFrench    \FBGlobalLayoutFrenchtrue
1123 \newif\ifFBReduceListSpacing     \FBReduceListSpacingfalse
1124 \newif\ifFBListOldLayout         \FBListOldLayoutfalse
1125 \newif\ifFBCompactItemize        \FBCompactItemizefalse
1126 \newif\ifFBStandardItemizeEnv    \FBStandardItemizeEnvtrue
1127 \newif\ifFBStandardEnumerateEnv  \FBStandardEnumerateEnvtrue
1128 \newif\ifFBStandardItemLabels    \FBStandardItemLabeltrue
1129 \newif\ifFBStandardLists         \FBStandardListtrue
1130 \newif\ifFBIndentFirst           \FBIndentFirstfalse
1131 \newif\ifFBFrenchFootnotes       \FBFrenchFootnotesfalse
1132 \newif\ifFBAutoSpaceFootnotes    \FBAutoSpaceFootnotesfalse
1133 \newif\ifFBOriginalTypewriter    \FBOriginalTypewriterfalse
1134 \newif\ifFBThinColonSpace        \FBThinColonSpacefalse
1135 \newif\ifFBThinSpaceInFrenchNumbers \FBThinSpaceInFrenchNumbersfalse
1136 \newif\ifFBFrenchSuperscripts    \FBFrenchSuperscripttrue
1137 \newif\ifFBLowercaseSuperscripts \FBLowercaseSuperscripttrue
1138 \newif\ifFBPartNameFull          \FBPartNameFulltrue
1139 \newif\ifFBCustomiseFigTabCaptions \FBCustomiseFigTabCaptionsfalse
1140 \newif\ifFBOldFigTabCaptions     \FBOldFigTabCaptionsfalse
1141 \newif\ifFBSmallCapsFigTabCaptions \FBSmallCapsFigTabCaptiontrue
1142 \newif\ifFBSuppressWarning       \FBSuppressWarningfalse
1143 \newif\ifFBINGuillSpace          \FBINGuillSpacefalse
```

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

```
1144 \edef\FB@french{\CurrentOption}
1145 \AtEndOfPackage{%
1146   \ifx\bbl@main@language\FB@french
```

```

1147 \FBGlobalLayoutFrenchtrue
1148 \@ifclassloaded{beamer}%
1149   {\PackageInfo{frenchb.ldf}{%
1150     No list customisation for the beamer class,%
1151     \MessageBreak reported}}%
1152   {\FBReduceListSpacingtrue
1153     \FBStandardItemizeEnvfalse
1154     \FBStandardEnumerateEnvfalse
1155     \FBStandardItemLabelsfalse}%
1156 \FBIndentFirsttrue
1157 \FBFrenchFootnotesttrue
1158 \FBAutoSpaceFootnotesttrue
1159 \FBCustomiseFigTabCaptionstrue
1160 \else
1161   \FBGlobalLayoutFrenchfalse
1162 \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.

```

1163 \RequirePackage{keyval}%
1164 \define@key{FB}{ShowOptions}[true]%
1165   {\csname FBShowOptions#1\endcsname}%
1166 \define@key{FB}{StandardLayout}[true]%
1167   {\csname FBStandardLayout#1\endcsname
1168     \ifFBStandardLayout
1169       \FBReduceListSpacingfalse
1170       \FBStandardItemizeEnvtrue
1171       \FBStandardItemLabelstrue
1172       \FBStandardEnumerateEnvtrue
1173       \FBIndentFirstfalse
1174       \FBFrenchFootnotesfalse
1175       \FBAutoSpaceFootnotesfalse
1176       \FBGlobalLayoutFrenchfalse
1177     \else
1178       \FBReduceListSpacingtrue
1179       \FBStandardItemizeEnvfalse
1180       \FBStandardItemLabelsfalse
1181       \FBStandardEnumerateEnvfalse
1182       \FBIndentFirsttrue
1183       \FBFrenchFootnotesttrue
1184       \FBAutoSpaceFootnotesttrue
1185     \fi}%
1186 \define@key{FB}{GlobalLayoutFrench}[true]%
1187   {\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.

```

1188   \ifFBGlobalLayoutFrench
1189     \ifx\babel@main@language\FB@french

```



```

1190             \else
1191                 \PackageWarning{frenchb.ldf}%
1192                     {Option 'GlobalLayoutFrench' skipped:%
1193                     \MessageBreak French is *not*
1194                     babel's last option.\MessageBreak}%
1195             \fi
1196         \fi}%
1197 \define@key{FB}{ReduceListSpacing}[true]%
1198     {\csname FBReduceListSpacing#1\endcsname}%
1199 \define@key{FB}{ListOldLayout}[true]%
1200     {\csname FBListOldLayout#1\endcsname
1201     \ifFBListOldLayout
1202         \FBStandardEnumerateEnvtrue
1203         \renewcommand*{\FrenchLabelItem}{\textendash}%
1204     \fi}%
1205 \define@key{FB}{CompactItemize}[true]%
1206     {\csname FBCompactItemize#1\endcsname
1207     \ifFBCompactItemize
1208         \FBStandardItemizeEnvfalse
1209         \FBStandardEnumerateEnvfalse
1210     \else
1211         \FBStandardItemizeEnvtrue
1212         \FBStandardEnumerateEnvtrue
1213     \fi}%
1214 \define@key{FB}{StandardItemizeEnv}[true]%
1215     {\csname FBStandardItemizeEnv#1\endcsname}%
1216 \define@key{FB}{StandardEnumerateEnv}[true]%
1217     {\csname FBStandardEnumerateEnv#1\endcsname}%
1218 \define@key{FB}{StandardItemLabels}[true]%
1219     {\csname FBStandardItemLabels#1\endcsname}%
1220 \define@key{FB}{ItemLabels}{%
1221     \renewcommand*{\FrenchLabelItem}{#1}}%
1222 \define@key{FB}{ItemLabeli}{%
1223     \renewcommand*{\Frlabelitemi}{#1}}%
1224 \define@key{FB}{ItemLabelii}{%
1225     \renewcommand*{\Frlabelitemii}{#1}}%
1226 \define@key{FB}{ItemLabeliii}{%
1227     \renewcommand*{\Frlabelitemiii}{#1}}%
1228 \define@key{FB}{ItemLabeliv}{%
1229     \renewcommand*{\Frlabelitemiv}{#1}}%
1230 \define@key{FB}{StandardLists}[true]%
1231     {\csname FBStandardLists#1\endcsname
1232     \ifFBStandardLists
1233         \FBReduceListSpacingfalse
1234         \FBCompactItemizefalse
1235         \FBStandardItemizeEnvtrue
1236         \FBStandardEnumerateEnvtrue
1237         \FBStandardItemLabelstrue
1238     \else
1239         \FBReduceListSpacingtrue
1240         \FBCompactItemizetrue

```

```

1241             \FBStandardItemizeEnvfalse
1242             \FBStandardEnumerateEnvfalse
1243             \FBStandardItemLabelsfalse
1244             \fi}%
1245 \define@key{FB}{IndentFirst}[true]%
1246             {\csname FBIndentFirst#1\endcsname}%
1247 \define@key{FB}{FrenchFootnotes}[true]%
1248             {\csname FBFrenchFootnotes#1\endcsname}%
1249 \define@key{FB}{AutoSpaceFootnotes}[true]%
1250             {\csname FBAutoSpaceFootnotes#1\endcsname}%
1251 \define@key{FB}{AutoSpacePunctuation}[true]%
1252             {\csname FBAutoSpacePunctuation#1\endcsname}%
1253 \define@key{FB}{OriginalTypewriter}[true]%
1254             {\csname FBOriginalTypewriter#1\endcsname}%
1255 \define@key{FB}{ThinColonSpace}[true]%
1256             {\csname FBThinColonSpace#1\endcsname}%
1257 \define@key{FB}{ThinSpaceInFrenchNumbers}[true]%
1258             {\csname FBThinSpaceInFrenchNumbers#1\endcsname}%
1259 \define@key{FB}{FrenchSuperscripts}[true]%
1260             {\csname FBFrenchSuperscripts#1\endcsname}%
1261 \define@key{FB}{LowercaseSuperscripts}[true]%
1262             {\csname FBLowercaseSuperscripts#1\endcsname}%
1263 \define@key{FB}{PartNameFull}[true]%
1264             {\csname FBPartNameFull#1\endcsname}%
1265 \define@key{FB}{CustomiseFigTabCaptions}[true]%
1266             {\csname FBCustomiseFigTabCaptions#1\endcsname}%
1267 \define@key{FB}{OldFigTabCaptions}[true]%
1268             {\csname FBOldFigTabCaptions#1\endcsname}%
    \CurrentOption no longer defined. It's value has been saved in \FB@CurOpt while
    reading frenchb.ldf.

1269             \ifFBOldFigTabCaptions
1270             \FB@addto{extras}{\babel@save\FBCaption@Separator
1271             \def\FBCaption@Separator{\CaptionSeparator}}}%
1272             \fi}%
1273 \define@key{FB}{SmallCapsFigTabCaptions}[true]%
1274             {\csname FBSmallCapsFigTabCaptions#1\endcsname}%
1275             \ifFBSmallCapsFigTabCaptions
1276             \let\FBfigtabshape\scshape
1277             \else
1278             \let\FBfigtabshape\relax
1279             \fi}%
1280 \define@key{FB}{SuppressWarning}[true]%
1281             {\csname FBSuppressWarning#1\endcsname}%
1282             \ifFBSuppressWarning
1283             \renewcommand{\FBWarning}[2]{\relax}%
1284             \fi}%

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

1285 \define@key{FB}{INGuillSpace}[true]%
1286             {\csname FBINGuillSpace#1\endcsname}%

```

```

1287 \define@key{FB}{InnerGuillSingle}[true]%
1288         {\csname FInnerGuillSingle#1\endcsname}%
1289 \define@key{FB}{EveryParGuill}{\expandafter\let\expandafter
1290         \FBeveryparguill\csname FBguill#1\endcsname}%
1291 \define@key{FB}{EveryLineGuill}{\expandafter\let\expandafter
1292         \FBeverylineguill\csname FBguill#1\endcsname
1293         \ifFB@luatex@punct
1294         \else
1295         \let\FBeverylineguill\FBguillnone
1296         \PackageWarning{frenchb.ldb}%
1297         {Option 'EveryLineGuill' skipped:%
1298         \MessageBreak this option is for
1299         LuaTeX *only*.\MessageBreak Reported}%
1300 \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@addGUILspace` attribute for LuaTeX or set `\XeTeXcharclass` of quotes to the proper value for XeTeX.

```

1301 \define@key{FB}{og}{%
1302     \ifFBunicode
1303         \ifFB@luatex@punct
1304         \FB@addGUILspace=1 \relax
1305     \fi

```

then with XeTeX it is a bit more tricky:

```

1306     \ifFB@xetex@punct

```

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

```

1307         \XeTeXcharclass"13 = \FB@guilo
1308         \XeTeXcharclass"AB = \FB@guilo
1309         \XeTeXcharclass"A0 = \FB@guilnul
1310         \XeTeXcharclass"202F = \FB@guilnul
1311     \fi
1312 \else

```

This is for conventional TeX engines:

```

1313         \newcommand*{\FB@og}{%
1314             \iflanguage{french}%
1315                 {\ifFBAutoSpaceGuill\FB@og\ignorespaces
1316                 \else\guillemotleft
1317                 \fi}%
1318             {\guillemotleft}}%
1319         \AtBeginDocument{%
1320             \ifdefined\DeclareInputText
1321             \ifdefined\uc@dccl
1322                 Package inputenc with utf8x encoding loaded, use \uc@dccl,
1323                 \uc@dccl{171}{default}{\FB@og}%
1324             \else
1325                 if encoding is not utf8x, try utf8...
1326                 \ifdefined\DeclareUnicodeCharacter
1327                     utf8 loaded, use \DeclareUnicodeCharacter,
1328                     \DeclareUnicodeCharacter{00AB}{\FB@og}%
1329                 \else
1330                     if utf8 is not loaded either, we assume 8-bit character input encoding. Package
1331                     MULEenc (from CJK) defines \mule@def to map characters to control sequences.
1332                     \@tempcnta'#1\relax
1333                     \ifdefined\mule@def
1334                         \mule@def{11}{\FB@og}%
1335                     \else
1336                         \DeclareInputText{\the\@tempcnta}{\FB@og}%
1337                     \fi
1338                 \fi
1339             \fi
1340         \else
1341             Package inputenc not loaded, no way...
1342             \PackageWarning{frenchb.ldf}%
1343             {Option 'og' requires package inputenc.\MessageBreak}%
1344         \fi
1345     }%
1346 \fi
1347 }%
1348 }%
1349 }%
1350 }%
1351 }%
1352 }%

Same code for the closing quote.
1342 \define@key{FB}{fg}{%
1343     \ifFBunicode
1344         \ifFB@luatex@punct
1345             \FB@addGUILspace=1 \relax
1346         \fi
1347         \ifFB@xetex@punct
1348             \XeTeXcharclass"14 = \FB@guilf
1349             \XeTeXcharclass"BB = \FB@guilf
1350             \XeTeXcharclass"A0 = \FB@guilnul
1351             \XeTeXcharclass"202F = \FB@guilnul
1352         \fi

```

```

1353     \else
1354         \newcommand*{\FB@@fg}{%
1355             \iflanguage{french}%
1356                 {\iffBAutoSpaceGuill\FB@fg
1357                 \else\guillemotright
1358                 \fi}%
1359             {\guillemotright}}%
1360     \AtBeginDocument{%
1361         \ifdefined\DeclareInputText
1362             \ifdefined\uc@dc\c
1363                 \uc@dc\c{187}{default}{\FB@@fg}%
1364             \else
1365                 \ifdefined\DeclareUnicodeCharacter
1366                     \DeclareUnicodeCharacter{00BB}{\FB@@fg}%
1367                 \else
1368                     \@tempcnta'#1\relax
1369                     \ifdefined\mule@def
1370                         \mule@def{27}{\FB@@fg}%
1371                     \else
1372                         \DeclareInputText{\the\@tempcnta}{\FB@@fg}%
1373                     \fi
1374                 \fi
1375             \fi
1376         \else
1377             \PackageWarning{frenchb.ldf}%
1378                 {Option 'fg' requires package inputenc.\MessageBreak}%
1379         \fi
1380     }%
1381 \fi
1382 }%
1383 }

```

\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.

```

1384 \newcommand*{\FBprocess@options}{%

```

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

```

1385     \ifpackageloaded{enumitem}{%
1386         \ifFBStandardItemizeEnv
1387         \else
1388             \FBStandardItemizeEnvtrue
1389             \PackageInfo{frenchb.ldf}%
1390                 {Setting StandardItemizeEnv=true for\MessageBreak
1391                 compatibility with enumitem package,\MessageBreak}%
1392         \fi
1393     \ifFBStandardEnumerateEnv

```

```

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

```

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

```

1423 \def\FB@ufl{\update@frenchlists}
1424 \ifx\bbl@main@language\FB@french
1425 \update@frenchlists
1426 \fi

```

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

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

```

1427 \ifBFAutoSpacePunctuation
1428 \autospace@beforeFDP
1429 \else
1430 \noautospace@beforeFDP
1431 \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.

```

1432 \ifFBOriginalTypewriter
1433 \else
1434     \let\ttfamilyORI\ttfamily
1435     \let\rmfamilyORI\rmfamily
1436     \let\sffamilyORI\sffamily
1437     \let\ttfamily\ttfamilyFB
1438     \let\rmfamily\rmfamilyFB
1439     \let\sffamily\sffamilyFB
1440 \fi

```

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

```

1441 \ifFBThinColonSpace
1442 \ifFB@luatex@punct
1443     \FBcolonskip=\FBthinskip\relax
1444 \else
1445     \renewcommand*{\FBcolonspace}{\FBthinspace}%
1446 \fi
1447 \fi

```

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

```

1448 \ifFBINGuillSpace
1449 \ifFB@luatex@punct
1450     \FBguillskip=3.33pt plus 1.665pt minus 1.11pt \relax
1451 \else
1452     \renewcommand*{\FBguillspace}{\space}%
1453 \fi
1454 \fi

```

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

```

1455 \@ifpackageloaded{numprint}%
1456 {\ifnprt@autolanguage
1457     \providecommand*{\npstylefrench}{}%
1458     \ifFBThinSpaceInFrenchNumbers
1459         \renewcommand*\npstylefrench{%
1460             \npthousandsep{,}%
1461             \npdecimalsign{,}%
1462             \npproductsign{\cdot}%
1463             \npunitseparator{,}%
1464             \npdegreeseparator{}%
1465             \nppercentseparator{\nprt@unitsep}%
1466         }%
1467     \else
1468         \renewcommand*\npstylefrench{%
1469             \npthousandsep{~}%
1470             \npdecimalsign{,}%
1471             \npproductsign{\cdot}%
1472             \npunitseparator{,}%

```

```

1473         \npdegreeseperator{}%
1474         \nppercentseparator{\nprt@unitsep}%
1475     }%
1476     \fi
1477     \npaddtolanguage{french}{french}%
1478     \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.

```

1479     \ifFBFrenchSuperscripts
1480         \DeclareRobustCommand*{\up}{\@ifstar{\FB@up@fake}{\fup}}%
1481     \else
1482         \DeclareRobustCommand*{\up}{\@ifstar{\FB@up@fake}%
1483                                         {\textsuperscript}}%
1484     \fi

```

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

```

1485     \ifFBLowercaseSuperscripts
1486     \else
1487         \renewcommand*{\FB@lc}[1]{##1}%
1488     \fi

```

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

```

1489     \ifFBCustomiseFigTabCaptions
1490         \ifFB@koma
1491             \renewcommand*{\captionformat}{\CaptionSeparator}%
1492         \fi
1493         \@ifclassloaded{memoir}%
1494             {\captiondelim{\CaptionSeparator}}{}%
1495         \@ifclassloaded{beamer}%
1496             {\defbeamertemplate{caption label separator}{FBcustom}{%
1497                 \CaptionSeparator}%
1498             \setbeamertemplate{caption label separator}[FBcustom]}{}%
1499     \else

```

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

```

1500         \ifFB@koma
1501             \renewcommand*{\captionformat}{\autospace@beforeFDP : }%
1502         \fi
1503         \@ifclassloaded{memoir}%
1504             {\captiondelim{\autospace@beforeFDP : }}%
1505             {}%
1506         \@ifclassloaded{beamer}%
1507             {\defbeamertemplate{caption label separator}{FBcolon}{%
1508                 \autospace@beforeFDP : }}%
1509             \setbeamertemplate{caption label separator}[FBcolon]%
1510             {}%
1511     \fi

```


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

```
1512 \ifFBShowOptions
1513 \GenericWarning{* }{%
1514   * **** List of possible options for frenchb ****\MessageBreak
1515   [Default values between brackets when frenchb is loaded *LAST*]%
1516   \MessageBreak
1517   ShowOptions=true [false]\MessageBreak
1518   StandardLayout=true [false]\MessageBreak
1519   GlobalLayoutFrench=false [true]\MessageBreak
1520   StandardLists=true [false]\MessageBreak
1521   IndentFirst=false [true]\MessageBreak
1522   ReduceListSpacing=false [true]\MessageBreak
1523   ListOldLayout=true [false]\MessageBreak
1524   StandardItemizeEnv=true [false]\MessageBreak
1525   StandardEnumerateEnv=true [false]\MessageBreak
1526   StandardItemLabels=true [false]\MessageBreak
1527   ItemLabels=\textendash, \textbullet,
1528   \protect\ding{43},... [\textendash]\MessageBreak
1529   ItemLabeli=\textendash, \textbullet,
1530   \protect\ding{43},... [\textendash]\MessageBreak
1531   ItemLabelii=\textendash, \textbullet,
1532   \protect\ding{43},... [\textendash]\MessageBreak
1533   ItemLabeliii=\textendash, \textbullet,
1534   \protect\ding{43},... [\textendash]\MessageBreak
1535   ItemLabeliv=\textendash, \textbullet,
1536   \protect\ding{43},... [\textendash]\MessageBreak
1537   FrenchFootnotes=false [true]\MessageBreak
1538   AutoSpaceFootnotes=false [true]\MessageBreak
1539   AutoSpacePunctuation=false [true]\MessageBreak
1540   OriginalTypewriter=true [false]\MessageBreak
1541   ThinColonSpace=true [false]\MessageBreak
1542   ThinSpaceInFrenchNumbers=true [false]\MessageBreak
1543   FrenchSuperscripts=false [true]\MessageBreak
1544   LowercaseSuperscripts=false [true]\MessageBreak
1545   PartNameFull=false [true]\MessageBreak
1546   SuppressWarning=true [false]\MessageBreak
1547   CustomiseFigTabCaptions=false [true]\MessageBreak
1548   OldFigTabCaptions=true [false]\MessageBreak
1549   SmallCapsFigTabCaptions=false [true]\MessageBreak
1550   INGullSpace=true [false]\MessageBreak
1551   InnerGullSingle=true [false]\MessageBreak
1552   EveryParGull=open, close, none [open]\MessageBreak
1553   EveryLineGull=open, close, none
1554   [open in LuaTeX, none otherwise]\MessageBreak
1555   og= <left quote character>, fg= <right quote character>%
1556   \MessageBreak
1557   *****%
1558   \MessageBreak\protect\frenchbsetup{ShowOptions}}
1559 \fi
1560 }
```

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.

```

1561 \AtBeginDocument{%
1562   \providecommand*\xspace{\relax}%

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

1563   \ifdefined\pdfstringdefDisableCommands
1564     \pdfstringdefDisableCommands{%
1565       \let\up\relax
1566       \let\up\relax
1567       \let\degre\textdegree
1568       \let\degres\textdegree
1569       \def\ieme{e\xspace}%
1570       \def\iemes{es\xspace}%
1571       \def\ier{er\xspace}%
1572       \def\iers{ers\xspace}%
1573       \def\iere{re\xspace}%
1574       \def\ieres{res\xspace}%
1575       \def\FrenchEnumerate#1{#1\degre\space}%
1576       \def\FrenchPopularEnumerate#1{#1\degre)\space}%
1577       \def\No{N\degre\space}%
1578       \def\no{n\degre\space}%
1579       \def\Nos{N\degre\space}%
1580       \def\nos{n\degre\space}%
1581       \def\FB@og{\guillemotleft\space}%
1582       \def\FB@fg{\space\guillemotright}%
1583       \def\at{@}%
1584       \def\circonflexe{\string^}%
1585       \def\tild{\string~}%
1586       \let\bsc\textsc
1587     }%
1588   \fi

```

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

```

1589   \FBprocess@options

```

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

```

1590   \ifFB@luatex@punct
1591     \activate@luatexpunct
1592   \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!

```

1593   \ifFBUnicode
1594     \ifdefined\DeclareUTFcharacter

```

```

1595     \else
1596         \@ifpackageloaded{luainputenc}{}%
1597             {\PackageWarning{frenchb.ldf}%
1598                 {Add \protect\usepackage{fontspec} to the\MessageBreak
1599                     preamble of your document,}%
1600             }%
1601     \fi
1602 \else
1603     \begingroup \newcommand{\FBOTone}{OT1}%
1604     \ifx\encodingdefault\FBOTone
1605         \PackageWarning{frenchb.ldf}%
1606             {OT1 encoding should not be used for French.%
1607             \MessageBreak
1608             Add \protect\usepackage[T1]{fontenc} to the
1609             preamble\MessageBreak of your document,}%
1610     \fi
1611 \endgroup
1612 \fi
1613 }

```

2.11 French lists

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

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

```

1614 \let\listORI\list
1615 \let\endlistORI\endlist
1616 \def\FB@listVsettings{%
1617     \setlength{\itemsep}{0.4ex plus 0.2ex minus 0.2ex}%
1618     \setlength{\parsep}{0.4ex plus 0.2ex minus 0.2ex}%
1619     \setlength{\topsep}{0.8ex plus 0.4ex minus 0.4ex}%
1620     \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`.

```

1621     \@tempdima=\parskip
1622     \addtolength{\topsep}{-\@tempdima}%
1623     \addtolength{\partopsep}{\@tempdima}%
1624 }

```

```

1625 \def\listFB#1#2{\listORI{#1}{\FB@listVsettings #2}}
1626 \let\endlistFB\endlist

```

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

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

```

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

```

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

```

\FrenchLabelItem \Frlabelitemi 1627 \newcommand*{\FrenchLabelItem}{\textemdash}
\FrenchLabelItem \Frlabelitemii 1628 \newcommand*{\Frlabelitemii}{\FrenchLabelItem}
\FrenchLabelItem \Frlabelitemiii 1629 \newcommand*{\Frlabelitemiii}{\FrenchLabelItem}
\FrenchLabelItem \Frlabelitemiv 1630 \newcommand*{\Frlabelitemiv}{\FrenchLabelItem}
\FrenchLabelItem 1631 \newcommand*{\Frlabelitemiv}{\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.

```

1632 \newlength\listindentFB
1633 \setlength{\listindentFB}{-1pt}
1634 \newlength\labelwidthFB
1635 \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.

```

1636 \newlength\leftmarginFB
1637 \def\FB@listHsettings{%
1638   \leftmarginFB\labelwidthFB
1639   \advance\leftmarginFB \labelsep
1640   \leftmargini\leftmarginFB
1641   \advance\leftmargini \listindentFB
1642   \leftmarginii\leftmarginFB
1643   \leftmarginiii\leftmarginFB

```

```

1644 \leftmarginiv\leftmarginFB
1645 \leftmargin\csname leftmargin\romannumeral\the\@listdepth\endcsname
1646 }

```

\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).

```

1647 \def\FB@itemizesettings{%
1648   \ifFBReduceListSpacing
1649     \setlength{\itemsep}{\z@}%
1650     \setlength{\parsep}{\z@}%
1651     \setlength{\topsep}{\z@}%
1652     \setlength{\partopsep}{\z@}%
1653     \@tempdima=\parskip
1654     \addtolength{\topsep}{-\@tempdima}%
1655     \addtolength{\partopsep}{\@tempdima}%
1656   \fi
1657   \settowidth{\labelwidth}{\csname\@itemitem\endcsname}%
1658   \ifFBListOldLayout
1659     \setlength{\leftmargin}{\labelwidth}%
1660     \addtolength{\leftmargin}{\labelsep}%
1661     \addtolength{\leftmargin}{\parindent}%
1662   \else
1663     \FB@listHsettings
1664   \fi
1665 }

```

The definition of \itemizeFB follows the one of \itemize in standard L^AT_EX 2_ε classes (see `ltlists.dtx`), spaces are customised by \FB@itemizesettings.

```

1666 \def\itemizeFB{%
1667   \ifnum \@itemdepth >\thr@@\toodeep\else
1668     \advance\@itemdepth\@ne
1669     \edef\@itemitem{\labelitem\romannumeral\the\@itemdepth}%
1670     \expandafter
1671     \listORI
1672     \csname\@itemitem\endcsname
1673     \FB@itemizesettings
1674   \fi
1675 }
1676 \let\enditemizeFB\endlistORI

1677 \def\labelitemsFB{%
1678   \let\labelitemi\Frlabelitemi
1679   \let\labelitemii\Frlabelitemii
1680   \let\labelitemiii\Frlabelitemiii
1681   \let\labelitemiv\Frlabelitemiv
1682   \ifdim\labelwidthFB<\z@
1683     \settowidth{\labelwidthFB}{\FrenchLabelItem}%
1684   \fi

```

```

1685 \ifdim\listindentFB<\z@
1686 \ifdim\parindent=\z@
1687 \setlength{\listindentFB}{1.5em}%
1688 \else
1689 \setlength{\listindentFB}{\parindent}%
1690 \fi
1691 \fi
1692 }

```

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

```

1693 \def\enumerateFB{%
1694 \ifnum \@enumdepth >\thr@@\@toodeep\else
1695 \advance\@enumdepth\@ne
1696 \edef\@enumctr{enum\romannumeral\the\@enumdepth}%
1697 \expandafter
1698 \list
1699 \csname label\@enumctr\endcsname
1700 {\FB@listHsettings
1701 \usecounter\@enumctr\def\makelabel##1{\hss\llap{##1}}}%
1702 \fi
1703 }
1704 \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).

```

1705 \def\descriptionFB{%
1706 \list{}{\FB@listHsettings
1707 \labelwidth\z@
1708 \itemindent-\leftmargin
1709 \ifnum\@listdepth=1
1710 \advance\itemindent by \listindentFB
1711 \fi
1712 \let\makelabel\descriptionlabel}%
1713 }
1714 \let\enddescriptionFB\endlistORI

```

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

```

\bbbl@frenchlistlayout 1715 \def\update@frenchlists{%
\bbbl@nonfrenchlistlayout 1716 \ifFBReduceListSpacing \let\list\listFB \fi
1717 \ifFBStandardItemizeEnv
1718 \else \let\itemize\itemizeFB \fi
1719 \ifFBStandardItemLabels
1720 \else \labelitemsFB \fi
1721 \ifFBStandardEnumerateEnv
1722 \else \let\enumerate\enumerateFB \let\description\descriptionFB \fi
1723 }

```

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

```

1724 \def\FB@ufl{\relax}
1725 \def\bbl@frenchlistlayout{%
1726   \ifFBGlobalLayoutFrench
1727   \else
1728     \babel@save\list          \babel@save\itemize
1729     \babel@save\enumerate     \babel@save\description
1730     \babel@save\labelitemi    \babel@save\labelitemii
1731     \babel@save\labelitemiii  \babel@save\labelitemiv
1732   \fi
1733   \FB@ufl
1734 }
1735 \def\bbl@nonfrenchlistlayout{%
1736   \ifFBGlobalLayoutFrench
1737     \update@frenchlists
1738   \fi
1739 }
1740 \FB@addto{extras}{\bbl@frenchlistlayout}
1741 \FB@addto{noextras}{\bbl@nonfrenchlistlayout}

```

2.12 French indentation of sections

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

```

1742 \def\bbl@frenchindent{%
1743   \ifFBGlobalLayoutFrench\else\babel@save\@afterindentfalse\fi
1744   \ifFBIndentFirst
1745     \let\@afterindentfalse\@afterindenttrue
1746     \@afterindenttrue
1747   \fi}
1748 \def\bbl@nonfrenchindent{%
1749   \ifFBGlobalLayoutFrench
1750     \ifFBIndentFirst
1751       \@afterindenttrue
1752     \fi
1753   \fi}
1754 \FB@addto{extras}{\bbl@frenchindent}
1755 \FB@addto{noextras}{\bbl@nonfrenchindent}

```

2.13 Formatting footnotes

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

The layout of footnotes is controlled by two flags `\ifFBAutoSpaceFootnotes` and `\ifFBFrenchFootnotes` which are set by options of `\frenchbsetup{}` (see section 2.10). The layout of footnotes *does not depend* on the current language (just think of two footnotes on the same page looking different because one was called in a French part, the other one in English!).

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

```

1756 \AtBeginDocument{\@ifpackageloaded{bigfoot}%
1757             {\PackageInfo{frenchb.ldb}%
1758             {bigfoot package in use.\MessageBreak
1759             frenchb will NOT customise footnotes;\MessageBreak
1760             reported}}%
1761             {\let\@footnotemarkORI\@footnotemark
1762             \def\@footnotemarkFB{\leavevmode\unskip\unkern
1763             \,\@footnotemarkORI}%
1764             \ifFBAutoSpaceFootnotes
1765             \let\@footnotemark\@footnotemarkFB
1766             \fi}%
1767     }

```

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).

```

1768 \newcommand*{\dotFFN}{.}
1769 \newcommand*{\kernFFN}{\kern .5em}
1770 \newdimen\parindentFFN
1771 \parindentFFN=10in
1772 \def\ftnISsymbol{\@fnsymbol\c@footnote}
1773 \long\def\@makefntextFB#1{\ifx\thefootnote\ftnISsymbol
1774     \@makefntextORI{#1}%
1775     \else
1776     \parindent=\parindentFFN
1777     \rule{z@}{\footnotesep}
1778     \setbox\@tempboxa\hbox{\@thefnmark}%
1779     \ifdim\wd\@tempboxa>z@
1780     \llap{\@thefnmark}\dotFFN\kernFFN
1781     \fi #1
1782     \fi}%

```

We save the standard definition of `\@makefntext` at the `\begin{document}`, and

then redefine `\@makefntext` according to the value of flag `\ifFBFrenchFootnotes` (true or false).

```

1783 \AtBeginDocument{\@ifpackageloaded{bigfoot}{}%
1784                 {\ifdim\parindentFFN<10in
1785                 \else
1786                     \parindentFFN=\parindent
1787                     \ifdim\parindentFFN<1.5em \parindentFFN=1.5em \fi
1788                 \fi
1789                 \let\@makefntextORI\@makefntext
1790                 \long\def\@makefntext#1{%
1791                     \ifFBFrenchFootnotes
1792                     \@makefntextFB{#1}%
1793                     \else
1794                     \@makefntextORI{#1}%
1795                     \fi}%
1796                 }%
1797             }

```

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

```

1798 \newcommand*{\AddThinSpaceBeforeFootnotes}{\FBAutoSpaceFootnotesttrue}
1799 \newcommand*{\FrenchFootnotes}{\FBFrenchFootnotesttrue}
1800 \newcommand*{\StandardFootnotes}{\FBFrenchFootnotesfalse}

```

2.14 Clean up and exit

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

```

1801 \FBclean@on@exit
1802 \let\FB@llc\loadlocalcfg
1803 \let\loadlocalcfg\@gobble
1804 \ldf@finish\CurrentOption
1805 \let\loadlocalcfg\FB@llc

```

3 Change History

v2.0	General: <code>\parindentFFN</code> not changed if already defined (required by JA for <code>cah-gut.cls</code>). 64	v2.0g	<code>\frenchbsetup</code> : Revert previous change to <code>StandardLayout</code> . This option must set the three flags <code>\FBReduceListSpacingfalse</code> , <code>\FBCompactItemizefalse</code> , and <code>\FBStandardItemLabeltrue</code> instead of <code>\FBStandardListstrue</code> , so that later options can still change their value before executing <code>\FBprocess@options</code> . Same thing for option <code>StandardLists</code> . . . 47
	Added warning for OT1 encoding. . 57		
	Footnotes are now printed by default ‘à la française’ for the whole document. 63	v2.1a	General: Command <code>\fup</code> added to produce better superscripts than <code>\textsuperscript</code> 35
	New command <code>\frenchbsetup</code> added for global customisation. . 46		<code>\datefrench</code> : <code>\today</code> changed (correction in 2.0 was wrong: <code>\today</code> was printed without spaces in toc). 34
	<code>\bsc</code> : <code>\hbox</code> dropped, replaced by <code>\kern0pt</code> 38		<code>\frenchbsetup</code> : New option: French-Superscripts to define <code>\up</code> as <code>\fup</code> or as <code>\textsuperscript</code> 47
	<code>\captionsfrench</code> : ‘Fig.’ changed to ‘Figure’ and ‘Tab.’ to ‘Table’. . . . 41		New option: LowercaseSuperscripts. 47
	<code>\datefrench</code> : 2 ‘relax’ added in <code>\today</code> ’s definition. 34	v2.1b	General: Disable some commands in bookmarks. 57
	<code>\FBtextellipsis</code> : Added special case for LY1 encoding, see bug report from Bruno Voisin (2004/05/18). 45		<code>\fup</code> : Command <code>\fup</code> changed to use real superscripts from fourier v. 1.6. 35
	<code>\nombre</code> : <code>\nombre</code> now requires <code>numprint.sty</code> 40	v2.1c	General: Added commands <code>\Nos</code> and <code>\nos</code> 37
v2.0b	General: Footnotes: Just do nothing (except warning) when the bigfoot package is loaded. 63		<code>\degres</code> : Provide a temporary definition (hyperref safe) of <code>\degres</code> in case it has to be expanded in the preamble (by beamer’s <code>\title</code> command for instance). 39
v2.0c	General: There is no need to define here <code>numprint</code> ’s command <code>\npstylefrench</code> , it will be re-defined ‘AtBeginDocument’ by <code>\FBprocess@options</code> 41		<code>\up</code> : Provide a temporary definition (hyperref safe) of <code>\up</code> in case it has to be expanded in the preamble (by beamer’s <code>\title</code> command for instance). 35
	<code>\frenchbsetup</code> : Option <code>ThinSpaceInFrenchNumbers</code> added. 47	v2.1d	General: Argument of <code>\ProvidesLanguage</code> changed above from ‘french’ to ‘frenchb’ (otherwise <code>\listfiles</code> prints no date/version information). The real name of current language
v2.0d	<code>\frenchbsetup</code> : Options <code>og</code> and <code>fg</code> changed: limit the definition to French so that quote characters can be used in German. 47		
v2.0e	<code>\frenchbsetup</code> : New option: <code>StandardLists</code> 47		
v2.0f	<code>\frenchbsetup</code> : <code>StandardLayout</code> option had no effect on lists. Test moved to <code>\FBprocess@options</code> . 47		
	Two typos corrected in option <code>StandardLists</code> : <code>[false] → [true]</code> and <code>StandardLayout → StandardLists</code> . 47		

(french) as to be corrected before calling \LdfInit.	13	v2.3c	\ttfamilyFB: Commands \ttfamily, \rmfamily and \sffamily have to be robust. Bug introduced in 2.3a, pointed out by Manuel Pégourié-Gonnard.	29
Avoid warning “\end occurred when \ifx ... incomplete” with LaTeX-2.09.	13			
v2.2a		v2.3d	\bbl@nonfrenchindent: Bug correction: previous versions of frenchb set the flag \if@afterindent to false outside French which is correct for English but wrong for some languages like Spanish. Pointed out by Juan José Torrens.	63
\frenchbsetup: Default values of flags changed: default now means ‘StandardLayout’, they will be changed to ‘FrenchLayout’ AtEndOfPackage only if french is \bbl@main@language.	47	v2.3e	\NoAutoSpaceBeforeFDP: Execute \AutoSpaceBeforeFDP also in LaTeX to define \FDP@colonspace: needed for tex4ht, pointed out by MPG.	29
The global layout of the document is no longer changed when frenchb is not the last option of babel (\bbl@main@language). Suggested by Ulrike Fischer.	47	v2.4a	General: \PackageWarning changed to \FBWarning (when bigfoot package in use).	64
When frenchb is babel’s last option, French becomes the document’s main language, so GlobalLayout-French applies.	47	\CaptionSeparator: \PackageWarning changed to \FBWarning (in case \@makecaption has been customised). \FBWarning is defined as \PackageWarning by default but can be made silent using \frenchbsetup, (suggested by MPG).	43	
\fup: \newif and \newdimen moved before \ifLaTeXe to avoid an error with plainTeX.	35	\frenchbsetup: New option SuppressWarning.	47	
v2.3a		\ifFBXeTeX: Added a new ‘if’ \FBunicode and some \lccode definitions to \extrasfrench and \noextrasfrench.	15	
General: In LaTeX, frenchb no longer adds spaces before ‘high punctuation’ characters in computer code. Suggested by Yannis Haralambous.	29	v2.4c	General: In \ttfamilyFB, also cancel automatic spaces inside French guillemets coded as characters (see \frenchbsetup).	51
\frenchbsetup: New option: OriginalTypewriter. Now frenchb switches to \noautospace@beforeFDP when a tt-font is in use. When OriginalTypewriter is set to true, frenchb behaves as in pre-2.3 versions.	47	\ttfamilyFB: In \ttfamilyFB, also cancel automatic spaces inside French guillemets entered as characters (see \frenchbsetup). ...	29	
\fup: \lowercase changed to \MakeLowercase as the former doesn’t work for non ASCII characters in encodings like applemac, utf-8,...	35	v2.4d	\up: Command \up defined with \providecommand instead of \newcommand as \up may be de-	
\NoAutoSpaceBeforeFDP: \NoAutoSpaceBeforeFDP and \AutoSpaceBeforeFDP now set the flag \ifFBAutoSpacePunctuation accordingly (LaTeX only).	29			
v2.3b				
General: New commands \dotFFN and \kernFFN for more flexibility (suggested by JA).	64			

finned elsewhere (catalan.ldf). Bug pointed out by Felip Manyé i Ballester.	35	\FBXeTeX and \FBLuaTeX as XeTeX and behave differently regarding the status of the French “apostrophe”.	15
v2.5a		v2.5e	
General: \og and \fg do not print correctly in English when using XeTeX or LuaTeX, fixed by using \textquotedblleft and \textquotedblright defined above.	32	General: \pdfstringdefDisableCommands should redefine \FB@og and \FB@fg instead of \og and \fg so that it works also when quotes are entered as characters. Reported by Sébastien Gouezel.	57
New command \NoAutoSpacing, suggested by MPG.	30	v2.5f	
Punctuation is no longer made active with XeTeX-based engines.	16	General: Changed definitions of \at, \circflex, \tild, \boi and \deg for Unicode based engines.	38
\captionsfrench: \emph deleted in \seename and \alsoname to match what is done for the other languages. Suggested by Marc Baudoin.	41	\FBtextellipsis: Unicode fonts also provide a ready made character for \textellipsis, let’s just use it (reported by Maxime Chupin, 2011/06/04).	45
\FBthinspace: Define \FBthinspace for those who want to customise the width of the space before ; and co.	17	v2.5g	
\textquotedblright: Change \guillemotleft and \guillemotright definitions for Unicode and provide definitions for \textquotedblleft and \textquotedblright. Insures correct printing of quotes by \og and \fg in French and outside.	31	General: Redefine \deg, \degres \at \circflex and \tild for bookmarks. Add \fup also.	58
v2.5b		When \ifFB@xetex@punct is true, ‘og’ and ‘fg’ options now set XeTeX-charclasses of these characters to \FB@guilo and \FB@guilf. Otherwise French quotes behave as normal characters (their XeTeXcharclass is 0).	51
General: Do not use the test \iflanguage{french} to check whether French is the main language or not, as it might be erroneously positive when English is the main language and no hyphenation patterns are available for French. In this case \l@french and \l@english are 0. Pointed out by Günter Milde.	47	\FB@xetex@punct@french: XeTeX-charclass(es) for French quotes will be set to \FB@guilo and \FB@guilf by options ‘og’ and ‘fg’ in \frenchbsetup. French quotes should behave as normal characters by default in XeLaTeX as in LaTeX.	25
v2.5c		v2.5h	
General: The code meant for XeTeX also works for LuaTeX, we just need to change the test.	51	\degres: textcomp.sty has changed. The test about \M@TS1 is no longer relevant, let’s change it.	39
v2.5d		v2.5i	
General: Moved the \newcount command outside \ifFB@xetex@punct ... \fi (it broke Plain formats). ..	25	General: Temporary fix: as long as xeCJK.sty will not use \newXeTeXintercharclass to allocate its classes, we will have to define 3 fake classes.	25
\ifFBXeTeX: Added two new ‘if’		\FB@xetex@punct@french: xeCJK.sty	

\captionsfrench to avoid a conflict with papertex.cls which loads datetime.sty.	13	\descriptionFB: Add \listindentFB to \itemindent. Suggested by Denis Bitouzé.	62
\bbl@nonfrenchguillemets deleted, use \babel@save instead.	32	\extrasfrench: Take advantage of babel's \babel@savevariable to handle apostrophe's \lccode. ..	15
Added explicit \FBguillskip for LuaTeX.	31	\FBprocess@options: Changed option ThinColonSpace to make it work also with LuaTeX.	55
Definitions of \FB@og and \FB@fg now depend on punctuation handling (LuaTeX / XeTeX / active). .	31	With koma-script and memoir class, customise \captionformat and \captiondelim.	56
french.cfg will be loaded (if found) instead of frenchb.cfg. NO NEED for .cfg files in French anyway. ..	65	\FBthinskip: LuaTeX requires dimensions: two new skips \FBcolonskip and \FBthinskip. 17	
In Plain, provide a substitute for \PackageWarning and \PackageInfo.	14	\frenchbsetup: New options OldFigTabCaptions and CustomiseFigTabCaptions.	47
Merging of \captionsfrenchb, \captionsfrançais with \captionsfrench deleted in favor of new babel 3.9 syntax.	42	v3.0b	
More informative, less TeXnical warning about \@makecaption. .	44	General: frenchb.lua was not found by Lua function dofile (not kpathsea aware). Call function kpse.find_file first, as suggested by Paul Gaborit.	24
New flag \ifFB@luatex@punct for 'high punctuation' management with LuaTeX engines.	16	Require luatexbase with LaTeX in case fontspec has not been loaded before babel.	17
New handling of 'high punctuation' through callbacks with LuaTeX engines.	17	v3.0c	
No warning about \@makecaption for SMF classes. No warning either with LuaTeX or XeTeX engines. ..	44	General: Activate option StandardLists when beamer class is loaded. ..	47
Options processing completely reorganised, now \babel@save and \babel@savevariable are usable for French.	46	Changed \FBguill@spacing (internal) to \FBguillspace (public). .	31
Support for options frenchb, français, canadien, acadian changed. 13		frenchb requires babel-3.9i.	14
Test \ifXeTeX changed to \ifBUnicode and 'xltextra' changed to 'fontspec'.	58	frenchb.lua: null glues should not trigger space insertion before high punctuation. Bug pointed out by Benoit Rivet for the 'lstlisting' environment of the listings package. 21	
\CaptionSeparator: Remove \CaptionSeparatorORI, use \babel@save instead.	43	Just load luatexbase.sty instead of luaotfload.sty with plain formats. 17	
\captionsfrench: Take advantage of babel's \SetString commands for captionnames.	41	No need to define \l@french as \lang@french, babel.def (3.9j) takes care for this.	13
\datefrench: Take advantage of babel's \SetString commands for \datefrench. Doesn't work with Plain (yet?).	34	\datefrench: \SetString still does not work for Plain with babel 3.9k. Need to define \datefrench. ...	34
		\frenchbsetup: New option INGuillSpace.	47
		v3.1a	
		General: Codes "13 and "14 added for French quotes in T1-encoding. Sup-	

port for older versions of LuaTeX and XeTeX dropped.	51	of SmallCapsFigTabCaptions. Pointed out by Céline Chevalier. .	47
fontspec is not required for T1 fonts used with the luainputenc.sty package.	58	v3.1f	
frenchb.lua: added flag addgl which must also be true when prev or next is not a char (i.e. kern0 in «\texttt{a}»).	22	General: \FBCaption@Separator changed when option CustomiseFigTabCaptions is set to false. ..	44
frenchb.lua: codes 0x13 and 0x14 added for French quotes in T1-encoding.	18	\FBprocess@options: Bug fix for the beamer class: figure and table captions are now consistent with frenchb's documentation. Pointed out by Denis Bitouzé.	56
frenchb.lua: look ahead when next is a kern (i.e. in « \texttt{a} »). ..	22	Definition of \captionformat and \captiondelim changed when option CustomiseFigTabCaptions is set to false.	56
Misplaced \fi for plain formats. ..	17	\FBthinspace: \FBthinspace is no longer a kern but a skip (frenchb adds a nobreak penalty before it). ..	17
New command \frquote for imbedded or long French quotations. ..	32	v3.1g	
\frenchbsetup: New options InnerGuillSingle, EveryParGuill and EveryLineGuill to control \frquote. ..	47	General: frenchb.lua: flag addgl set to false for '«' at the end of an \hbox or a paragraph or when followed by a null glue (i.e. springs).	22
v3.1b		frenchb.lua: flag addgl set to false for '»' at the beginning of an \hbox or a paragraph or a tabular 'l' and 'c' columns.	22
General: frenchb.lua: add a check for null fid in french_punctuation (Tikz \nullfont). Bug pointed out by Paul Gaborit.	20	frenchb.lua: node HLIST added; node TEMP added for the first node of \hboxes.	19
\captionsfrench: Change \scshape to customisable \FBfigtabshape for \figurename and \tablename.	41	Lua function french_punctuation is now inserted at the end of the "kerning" callback (no priority) instead of "hpack_filter" and "pre_linebreak_filter".	24
\fprimo): Removed \lowercase from definitions of \FrenchEnumerate, ... \No and co: \up already does the conversion.	37	Use Babel defined loops \bbl@for instead of \@for borrowed from file ltcntrl.dtx (\@for is undefined in Plain).	25
\frenchbsetup: New option SmallCapsFigTabCaptions.	47	\captionsfrench: \partname's definition depends now on flag PartNameFull. No need to redefine it in \frenchbsetup.	41
\ieres: Removed \lowercase from definitions of \ieme and co: \up already does the conversion. ...	37	Bug fix for koma-scripts classes: a spurious dot was added by the \partformat command.	42
v3.1c		\frenchbsetup: PartNameFull now just sets the flag, nothing to add to \captionsfrench when false. ..	47
General: frenchb.lua: Previous bug fix for null glues (v3.0c) did not work properly. Fixed now (I hope). Pointed out by Jacques André. ..	21	v3.1h	
v3.1d		General: french.cfg from e-french con-	
General: New section: issue warnings if packages listings, numprint and natbib are loaded too early or too late vs babel.	46		
v3.1e			
\frenchbsetup: Corrected typo: SmallCapsFigTabCaptions instead			

flits with frenchb. Do NOT load it (no need for .cfg files with frenchb anyway).	65	v3.1k	General: (pdfTeX shorthands) test on \lastskip changed from 0pt to 1sp for active punctuation for con- sistency with XeTeX and LuaTeX.	27
v3.1i			\FB@xetex@punct@french: Thin glues (less than 1sp) should not trigger space insertion before high ponc- tuation. Add a check on \lastkip.	25
General: \nombre command changed when numprint.sty is not loaded: only one warning, no error.	40			
Compatibility code added due to changes in the 2015/10/01 LaTeX release, see ltnews23.tex.	17	v3.1l	General: Add a variant of \babel@savevariable to save \XeTeXcharclass(es) in a loop.	25
Remove restriction about loading numprint.sty after babel.	46		frenchb.lua: font.getfont(fid) pos- sibly returns nil even for a posi- tive fid (i.e. AMS lcircle1.pfb). Re- ported by François Legendre.	20
\frquote: \luatexlocalleftbox changed to \localleftbox by new LaTeX release 2015/10/01.	33		\FB@luatex@punct@french: Use \babel@save to save and restore \shorthandon and \shorthandoff.	23
v3.1j			\FB@xetex@punct@french: Save and restore \XeTeXinterchartokenstate, \shorthandon, \shorthandoff using \babel@savevariable and \babel@save, \XeTeXcharclass(es) using \FB@savevariable@loop.	25
General: Loading luatexbase.sty is no longer needed with LaTeX release 2015/10/01 or later.	17			
\frquote: \PackageWarning is unde- fined in Plain, use \fb@warning in- stead.	33			
\fr@quote completely rewritten: \leavevmode added and explic- itly save/retore \everypar and \localleftbox instead of using a group in order to ensure compati- bility with package wrapfig.	33			