

A Babel language definition file for French

frenchb.dtx v3.0b, 2014/03/14

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	7
1.3 Hyphenation checks	7
1.4 Changes	8
2 The code	12
2.1 Initial setup	12
2.2 Punctuation	15
2.2.1 Punctuation with LuaTeX	16
2.2.2 Punctuation with XeTeX	22
2.2.3 Punctuation with standard (pdf)TeX	25
2.2.4 Punctuation switches common to all engines	26
2.3 Commands for French quotation marks	28
2.4 Date in French	30
2.5 Extra utilities	31
2.6 Formatting numbers	35
2.7 Caption names	36
2.8 Dots...	40
2.9 Setup options: keyval stuff	41
2.10 French lists	53
2.11 French indentation of sections	57
2.12 Formatting footnotes	57
2.13 Clean up and exit	59
3 Change History	60

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.

`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 v3.0b are listed in subsection 1.4 p. 8.

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. 7.

¹The file described in this section has version number v3.0b and was last revised on 2014/03/14.

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

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

1. French hyphenation patterns are made active;
2. ‘high punctuation’ characters (: ; ! ?) automatically add correct spacing in French; this is achieved using callbacks in Lua(La)TeX or ‘XeTeXinter-char’ mechanism in Xe(La)TeX; with TeX’82 and pdf(La)TeX these four characters are made active in the whole document;
3. `\today` prints the date in French;
4. the caption names are translated into French (L^AT_EX only). For customisation of caption names see section 1.2.2 p. 7.
5. the space after `\dots` is removed in French.

Some commands are provided in `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 marks~>>`: `\og` and `\fg` provide better horizontal spacing. `\og` and `\fg` can be used outside French, they typeset then English quotes “ and ”.
2. A command `\up` is provided to typeset superscripts like `M\up{me}` (abbreviation for “Madame”), `l\up{er}` (for “premier”). Other commands are also provided for ordinals: `\ier`, `\iere`, `\iers`, `\ieres`, `\ieme`, `\iemes` (`3\iemes` prints 3^{es}).
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°, 2°, 3°, 4°. `\FrenchEnumerate{6}` prints 6°.
5. Abbreviations for “Numéro(s)” and “numéro(s)” (N° N^{os} n° and n^{os}) are obtained via the commands `\No`, `\Nos`, `\no`, `\nos`.
6. Two commands are provided to typeset the symbol for “degré”: `\degre` prints the raw character and `\degres` should be used to typeset temperatures (e.g., “20~\degres C” with an unbreakable space), or for alcohols’ strengths (e.g., “45\degres” with *no* space in French).
7. In math mode the comma has to be surrounded with braces to avoid a spurious space being inserted after it, in decimal numbers for instance (see the T_EXbook p. 134). The command `\DecimalMathComma` makes the comma be an ordinary character *in French only* (no space added); as a

³ `\selectlanguage{français}` and `\selectlanguage{frenchb}` are kept for backward compatibility but should no longer be used.

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 `'\ier juin'` will print `'1er juin'` (no need for a forced space after `\ier`).

1.2 Customisation

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

1.2.1 `\frenchbsetup{<options>}`

`\frenchbsetup{ShowOptions}` prints all available options to the `.log` file, it is just meant as a remainder of the list of offered options. As usual with `keyval` syntax, boolean options (as `ShowOptions`) can be entered as `ShowOptions=true` or just `ShowOptions`, the `=true` part can be omitted.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

1.2.2 Captions

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

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

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

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

1.3 Hyphenation checks

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

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

%%% Test file for French hyphenation.

```

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

```

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

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

Frequent mismatches:

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

1.4 Changes

What’s new in version 3.0?

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.7.
- \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.

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 unbreable 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 $\text{\LaTeX}2_{\epsilon}$ in compatibility mode has been dropped. This version is meant for $\text{\LaTeX}2_{\epsilon}$ and Plain based formats (like `bplain`). $\text{\LaTeX}2_{\epsilon}$ formats based on `ml\TeX` are no longer supported either (plenty of good 8-bits fonts are available now, so T1 encoding should be preferred for typesetting in French). A warning is issued when OT1 encoding is in use at the `\begin{document}`.
- Customisation should now be handled only by command `\frenchbsetup{}`, `frenchb.cfg` (kept for compatibility) should no longer be used. See section 1.2 for the list of available options.
- Captions in figures and tables have changed in French: former abbreviations “Fig.” and “Tab.” have been replaced by full names “Figure” and “Table”. If this leads to formatting problems in captions, you can add the following two commands to your preamble (after loading `babel`) to get the

former captions

```
\addto\captionsfrench{\def\figurename{{\scshape Fig.}}}
\addto\captionsfrench{\def\tablename{{\scshape Tab.}}}.
```

- The `\nombre` command is now provided by the `numprint` package which has to be loaded *after* `babel` with the option `autolanguage` if number formatting should depend on the current language.
- The `\bsc` command no longer uses an `\hbox` to stop hyphenation of names but a `\kern0pt` instead. This change enables `microtype` to fine tune the length of the argument of `\bsc`; as a side-effect, compound names like Dupont-Durand can now be hyphenated on explicit hyphens. You can get back to the former behaviour of `\bsc` by adding

```
\renewcommand*{\bsc}[1]{\leavevmode\hbox{\scshape #1}}
```

to the preamble of your document.
- Footnotes are now displayed “à la française” for the whole document, except with an explicit

```
\frenchbsetup{AutoSpaceFootnotes=false,FrenchFootnotes=false}.
```

Add this command if you want standard footnotes. It is still possible to revert locally to the standard layout of footnotes by adding `\StandardFootnotes` (inside a `minipage` environment for instance).

2 The code

2.1 Initial setup

First make sure that `\l@french` is defined (possibly as 0). In Plain formats, `@catcode` is not ‘letter’; LuaTeX and XeTeX formats set `\lang@<language>` instead of `\l@<language>`.

```
1 \chardef\atcatcode=\catcode'\@
2 \catcode'\@=11\relax
3 \def\FB@nopatterns{%
4   \ifx\l@nohyphenation\@undefined
5     \edef\bbl@nulllanguage{\string\language=0}%
6     \adddialect\l@french0
7   \else
8     \adddialect\l@french\l@nohyphenation
9     \edef\bbl@nulllanguage{\string\language=nohyphenation}%
10  \fi
11  \@nopatterns{French}}
12 \ifx\l@french\@undefined
13   \ifx\lang@french\@undefined
14     \FB@nopatterns
15   \else
16     \ifnum\lang@french>0
17       \let\l@french\lang@french
18     \else
19       \FB@nopatterns{French}
20   \fi
21 \fi
22 \fi
```

If `frenchb.ldf` was loaded with babel’s options `francais` or `frenchb`, we make it behave as if `french` was specified.

```
23 \def\bbl@tempa{francais}
24 \ifx\CurrentOption\bbl@tempa
25   \let\l@francais\l@french
26   \def\captionsfrancais{\captionsfrench}
27   \def\datefrancais{\datefrench}
28   \def\extrasfrancais{\extrasfrench}
29   \def\noextrasfrancais{\extrasfrench}
30   \def\CurrentOption{french}
31 \fi
32 \def\bbl@tempa{frenchb}
33 \ifx\CurrentOption\bbl@tempa
34   \let\l@frenchb\l@french
35   \def\captionsfrenchb{\captionsfrench}
36   \def\datefrenchb{\datefrench}
37   \def\extrasfrenchb{\extrasfrench}
38   \def\noextrasfrenchb{\extrasfrench}
39   \def\CurrentOption{french}
40 \fi
41 \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.

```
42 \LdfInit\CurrentOption\captionsfrench
```

Quit if babel's version is less than 3.9.

```
43 \let\bbl@tempa\relax
44 \ifx\StartBabelCommands\@undefined
45   \let\bbl@tempa\endinput
46   \PackageError{frenchb.ldf}
47     {frenchb requires babel v.3.9.\MessageBreak
48       Aborting here}
49     {Please upgrade Babel!}
50 \fi
51 \bbl@tempa
```

`\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.

```
52 \newif\ifLaTeXe
53 \let\bbl@tempa\relax
54 \ifx\magnification\@undefined
55   \ifx\@compatibilitytrue\@undefined
56     \PackageError{frenchb.ldf}
57       {LaTeX-2.09 format is no longer supported.\MessageBreak
58         Aborting here}
59       {Please upgrade to LaTeX2e!}
60     \let\bbl@tempa\endinput
61   \else
62     \LaTeXettrue
63   \fi
64 \fi
65 \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.

```
66 \ifx\l@acadian\@undefined
67   \ifx\l@canadien\@undefined
68     \adddialect\l@acadian\l@french
69     \adddialect\l@canadien\l@french
70   \else
71     \adddialect\l@acadian\l@canadien
72   \fi
73 \else
74   \adddialect\l@canadien\l@acadian
75 \fi
76 \def\bbl@tempa{canadien}
77 \ifx\CurrentOption\bbl@tempa
78   \def\captionscanadien{\captionacadian}
```

```

79 \def\datecanadien{\dateacadian}
80 \def\extrascanadien{\extrasacadian}
81 \def\noextrascanadien{\extrasacadian}
82 \def\CurrentOption{acadian}
83 \fi

```

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

```

84 \def\fb@warning#1{%
85   \begingroup
86     \newlinechar='\^^J
87     \def\{\^^J(frenchb.ldf) }%
88     \message{\#1}%
89   \endgroup}
90 \def\fb@info#1{%
91   \begingroup
92     \newlinechar='\^^J
93     \def\{\^^J}%
94     \wlog{\#1}%
95   \endgroup}

```

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

```

96 \expandafter\providehyphenmins\expandafter{\CurrentOption}{\tw@\thr@@}

```

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

\ifFBLuaTeX XeTeX 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.

```

97 \newif\ifFBUnicode
98 \newif\ifFBLuaTeX
99 \newif\ifFBXeTeX
100 \begingroup\expandafter\expandafter\expandafter\endgroup
101 \expandafter\ifx\csname luatexversion\endcsname\relax
102 \else
103   \FBUnicodetrue \FBLuaTeXtrue
104 \fi
105 \begingroup\expandafter\expandafter\expandafter\endgroup
106 \expandafter\ifx\csname XeTeXrevision\endcsname\relax
107 \else
108   \FBUnicodetrue \FBXeTeXtrue
109 \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.

```

110 \@namedef{extras\CurrentOption}{%
111     \babel@savevariable{\lccode'\'}%
112     \ifFBunicode
113         \babel@savevariable{\lccode"2019}%
114         \lccode'\']="2019\lccode"2019="2019
115     \else
116         \lccode'\]='\'
117     \fi
118 }
119 \@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).

```

120 \let\FB@CurOpt\CurrentOption
121 \newcommand*{\FB@addto}[2]{%
122     \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.

```

123 \FB@addto{extras}{\bbl@frenchspacing}
124 \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:

```

125 \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).

```

126 \newif\ifFB@active@punct \FB@active@puncttrue
127 \newif\ifFB@luatex@punct
128 \ifBLuaTeX
129     \ifnum\luatexversion>75
130         \FB@luatex@puncttrue\FB@active@punctfalse

```

```

131 \fi
132 \fi

```

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

```

133 \newif\ifFB@xetex@punct
134 \begingroup\expandafter\expandafter\expandafter\endgroup
135 \expandafter\ifx\csname XeTeXinterchartokenstate\endcsname\relax
136 \else
137   \FB@xetex@puncttrue\FB@active@punctfalse
138 \fi

```

\FBcolonspace According to the I.N. specifications, the ‘:’ requires an inter-word space before it, **\FBthinspace** the other three require just a `\thinspace`. We define **\FBcolonspace** as `\space` **\FBcolonskip** (inter-word space) and **\FBthinspace** as `\thinspace` (both are user customisable). **\FBthinskip** (able). LuaTeX requires skips instead of commands, so we define **\FBcolonskip** and **\FBthinskip** to hold the specifications (width/stretch/shrink) of `\space` and `\thinspace` for the `lmr10` font; these parameters will be scaled for the current font by the `frenchb.lua` script (see how p. 18). **\FBcolonskip** and **\FBthinskip** are also user customisable.

```

139 \newcommand*{\FBcolonspace}{\space}
140 \newcommand*{\FBthinspace}{\thinspace}
141 \newskip\FBcolonskip
142 \FBcolonskip=3.33pt plus 1.665pt minus 1.11pt \relax
143 \newskip\FBthinskip
144 \FBthinskip=1.66672pt \relax

```

2.2.1 Punctuation with LuaTeX

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

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

```

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

```



```

162             with this version of LuaTeX!}
163     \fi
164 \fi
165 \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).

```

166 local FB_punct_thin =
167   {[string.byte("!")] = true,
168    [string.byte("?")] = true,
169    [string.byte(";")] = true}
170 local FB_punct_thick =
171   {[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.

```

172 local FB_punct_left =
173   {[string.byte("!")] = true,
174    [string.byte("?")] = true,
175    [string.byte(";")] = true,
176    [string.byte(":")] = true,
177    [0xBB] = true}
178 local FB_punct_right =
179   {[0xAB] = true}

```

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

```

180 local FB_punct_null =
181   {[string.byte("!")] = true,
182    [string.byte("?")] = true,
183    [string.byte("[")] = true,
184    [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.

```

185   [0xA0] = true,
186   [0x202F] = true}
187 local FB_guil_null =
188   {[0xA0] = true,
189    [0x202F] = true}

```

Local definitions for nodes:

```

190 local new_node = node.new
191 local copy_node = node.copy
192 local node_id = node.id
193 local GLUE = node_id("glue")
194 local GSPEC = node_id("glue_spec")
195 local GLYPH = node_id("glyph")
196 local PENALTY = node_id("penalty")

```

```

197 local nobreak      = new_node(PENALTY)
198 nobreak.penalty    = 10000
199 local insert_node_before = node.insert_before
200 local insert_node_after  = node.insert_after
201 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...

```

202 local thin10 = tex.skip['FBthinskip']
203 local thinwd = thin10.width/65536/3.33
204 local thinst = thin10.stretch/65536/1.665
205 local thinsh = thin10.shrink/65536/1.11
206 local coln10 = tex.skip['FBcolonskip']
207 local colnwd = coln10.width/65536/3.33
208 local colnst = coln10.stretch/65536/1.665
209 local colnsh = coln10.shrink/65536/1.11
210 local guil10 = tex.skip['FBguillskip']
211 local guilwd = guil10.width/65536/3.33
212 local guilst = guil10.stretch/65536/1.665
213 local guilsh = guil10.shrink/65536/1.11

```

and a function to scale them for the current font:

```

214 local font_table = {}
215 local function new_glue_scaled (fid,width,stretch,shrink)
216   local fp = font_table[fid]
217   if not fp then
218     font_table[fid] = font.getfont(fid).parameters
219     fp = font_table[fid]
220   end
221   local gl = new_node(GLUE,0)
222   local gl_spec = new_node(GSPEC)
223   gl_spec.width = width * fp.space
224   gl_spec.stretch = stretch * fp.space_stretch
225   gl_spec.shrink = shrink * fp.space_shrink
226   gl.spec = gl_spec
227   return gl
228 end

```

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

```

229 local addDPspace    = luatexbase.attributes['FB@addDPspace']
230 local addGUILspace  = luatexbase.attributes['FB@addGUILspace']
231 local has_attribute = node.has_attribute

```

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

```

232 local function french_punctuation (head)
233   for item in node.traverse_id(GLYPH, head) do
234     local lang = item.lang
235     local char = item.char
236     local SIG = has_attribute(item, addGUILspace)
237     if lang == FR and FB_punct_left[char] then
238       local fid = item.font
239       local prev = item.prev
240       local prev_id, prev_subtype, prev_char
241       if prev then
242         prev_id = prev.id
243         prev_subtype = prev.subtype
244         if prev_id == GLYPH then
245           prev_char = prev.char
246         end
247       end
248       local glue = prev_id == GLUE and prev_subtype == 0

```

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

```

249     if FB_punct_thin[char] or FB_punct_thick[char] then
250       local SBDP = has_attribute(item, addDPspace)
251       local fbg glue
252       if FB_punct_thick[char] then
253         fbg glue = new_glue_scaled(fid, colnwd, colnst, colnsh)
254       else
255         fbg glue = new_glue_scaled(fid, thinwd, thinst, thinsh)
256       end
257       local auto =
258         SBDP and SBDP > 0 and
259         (prev_char and not FB_punct_null[prev_char]) or
260         (not prev_char and (prev_id ~= 0 or prev_subtype ~= 3))
261       if glue or auto then
262         if glue then
263           head = remove_node(head, prev, true)
264         end
265         insert_node_before(head, item, copy_node(nobreak))
266         insert_node_before(head, item, copy_node(fbg glue))
267       end

```

Let's consider `'»'` now (the only remaining glyph of `FB_punct_left` class): we just have to remove any *glue* possibly preceeding `'»'`, then insert the nobreak penalty and the proper *glue* (controlled by `\FBguillskip`). This is done only if French quotes have been 'activated' by options `og=«`, `fg=»` in `\frenchbsetup{}` and can

be denied locally with \NoAutoSpacing (this is controlled by the SIG flag).

```

268         elseif SIG and SIG > 0 then
269             if glue or (prev_char and not FB_guil_null[prev_char]) then
270                 if glue then
271                     head = remove_node(head,prev,true)
272                 end
273                 local fbglue = new_glue_scaled(fid,guilwd,guilst,guilsh)
274                 insert_node_before(head, item, copy_node(nobreak))
275                 insert_node_before(head, item, copy_node(fbglue))
276             end
277         end
278     end

```

Similarly, for ‘«’ (unique member of the FB_punct_right class), we check the following node looking for *glue* or FB_guil_null character, in order to remove any *glue* possibly following it and to insert the proper *glue* and nobreak penalty in this order if necessary.

```

279     if lang == FR and FB_punct_right[char] and SIG and SIG > 0 then
280         local next = item.next
281         local next_id, next_subtype, next_char
282         if next then
283             next_id = next.id
284             next_subtype = next.subtype
285             if next_id == GLYPH then
286                 next_char = next.char
287             end
288         end
289         local glue = next_id == GLUE and next_subtype == 0
290         if glue or (next_char and not FB_guil_null[next_char]) then
291             if glue then
292                 head = remove_node(head,next,true)
293             end
294             local fid = item.font
295             local fbglue = new_glue_scaled(fid,guilwd,guilst,guilsh)
296             insert_node_after(head, item, copy_node(fbglue))
297             insert_node_after(head, item, copy_node(nobreak))
298         end
299     end
300 end
301 return head
302 end
303 return french_punctuation

```

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

```

304 \ifFB@luatex@punct
305   \newcommand*{\FB@luatex@punct@french}{%
306     \ifx\shorthandoffORI\@undefined
307       \let\shorthandonORI\shorthandon

```

```

308     \let\shorthandoffORI\shorthandoff
309   \fi
310   \def\shorthandoff##1{%
311     \ifx\PackageWarning\@undefined
312       \fb@warning{\noexpand\shorthandoff{;:!?} is helpless with
313         LuaTeX,\ \ use \noexpand\NoAutoSpacing
314         *inside a group* instead.}%
315     \else
316       \PackageWarning{frenchb.lda}{\protect\shorthandoff{;:!?} is
317         helpless with LuaTeX,\MessageBreak use \protect\NoAutoSpacing
318         \space *inside a group* instead;\MessageBreak reported}%
319     \fi}%
320   \def\shorthandon##1{%
321   }
322   \newcommand*{\FB@luatex@punct@nonfrench}{%
323     \ifx\shorthandoffORI\@undefined
324     \else
325       \let\shorthandon\shorthandonORI
326       \let\shorthandoff\shorthandoffORI
327     \fi
328   }
329   \FB@addto{extras}{\FB@luatex@punct@french}
330   \FB@addto{noextras}{\FB@luatex@punct@nonfrench}

```

In L^AT_EX 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 to both callbacks pre_linebreak_filter (paragraph building) and hpack_filter (\hbox building).

```

331   \def\activate@luatexpunct{%
332     \directlua{%
333       FR = \the\l@french
334       local path = kpse.find_file("frenchb.lua", "lua")
335       if path then
336         local f = dofile(path)
337         luatexbase.add_to_callback("pre_linebreak_filter",
338           f, "frenchb.french_punctuation",1)
339         luatexbase.add_to_callback("hpack_filter",
340           f, "frenchb.french_punctuation",1)
341       else
342         texio.write_nl('')
343         texio.write_nl('*****')
344         texio.write_nl('Error: frenchb.lua not found.')
345         texio.write_nl('*****')
346         texio.write_nl('')
347       end
348     }%
349   }
350 \fi

```

End of specific code for punctuation with LuaTeX engines.

2.2.2 Punctuation with XeTeX

If `\XeTeXinterchartokenstate` is available, we use the “inter char” mechanism to provide correct spacing in French before the four characters `; ! ?` and `:. The basis of the following code was borrowed from the polyglossia package, see gloss-french.ldf. We use the same mechanism for French quotes (« and »), when automatic spacing for quotes is required by options og= and fg= in \frenchbsetup{} (see section 2.9).`

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

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

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

```
351 \newcount\FB@interchartokenstateORI
352 \ifFB@xetex@punct
353   \ifLaTeXe
354     \PackageInfo{frenchb.ldf}{No need for active punctuation characters%
355       \MessageBreak with this version of XeTeX!%
356       \MessageBreak reported}
357   \else
358     \fb@info{No need for active punctuation characters\\
359       with this version of XeTeX!}
360   \fi
```

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

```
361 \ifx\@for\@undefined
362   \def\@nnil{\@nil}%
363   \def\@empty{}%
364   \def\@fornoop#1\@#2#3{%
365     \long\def\@for#1:=#2\do#3{%
366       \expandafter\def\expandafter\@fortmp\expandafter{#2}%
367       \ifx\@fortmp\@empty \else
368         \expandafter\@forloop#2,\@nil,\@nil\@#1{#3}\fi}%
369     \long\def\@forloop#1,#2,#3\@#4#5{\def#4{#1}\ifx #4\@nnil \else
370       #5\def#4{#2}\ifx #4\@nnil \else#5\@iforloop #3\@#4{#5}\fi\fi}%
371     \long\def\@iforloop#1,#2\@#3#4{\def#3{#1}\ifx #3\@nnil
372       \expandafter\@fornoop \else
373       #4\relax\expandafter\@iforloop\fi#2\@#3{#4}}%
374     \def\@tfor#1:={\@tfor#1 }%
375     \long\def\@tfor#1#2\do#3{\def\@fortmp{#2}\ifx\@fortmp\space\else
376       \@tforloop#2\@nil\@nil\@#1{#3}\fi}%
377     \long\def\@tforloop#1#2\@#3#4{\def#3{#1}\ifx #3\@nnil
378       \expandafter\@fornoop \else
379       #4\relax\expandafter\@tforloop\fi#2\@#3{#4}}%
380   \fi
```

Six new character classes are defined for `frenchb`.

```

381 \newXeTeXintercharclass\FB@punctthick
382 \newXeTeXintercharclass\FB@punctthin
383 \newXeTeXintercharclass\FB@punctnul
384 \newXeTeXintercharclass\FB@guilo
385 \newXeTeXintercharclass\FB@guilf
386 \newXeTeXintercharclass\FB@guilnul

```

We define a command to store the `\XeTeXcharclass` values which will be modified for French (as a comma separated list) and a command to retrieve them.

```

387 \def\FB@charclassesORI{}
388 \def\empty{}
389 \def\FB@parse#1,#2\endparse{\def\FB@class{#1}%
390 \def\FB@charclassesORI{#2}}%

```

`\FB@xetex@punct@french` The following command will be executed when entering French, it first saves the values to be modified, then fits them to our needs. It also redefines `\shorthandoff` and `\shorthandon` (locally) to avoid error messages with XeTeX-based engines.

```

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

```

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

```

392 \if\FB@charclassesORI\empty
393 \FB@interchartokenstateORI=\XeTeXinterchartokenstate
394 \@for\FB@char:={'\:,\;\,'!\,'?', "AB,"BB,%
395 '\(,\[, '\{,\,, '\., '\-, '\), '\], '\},%
396 '\%, "22,"27,"60,"2019,"A0,"202F}\do
397 {\edef\FB@charclassesORI{\FB@charclassesORI%
398 \theXeTeXcharclass\FB@char,}}%
399 \let\shorthandonORI\shorthandon
400 \let\shorthandoffORI\shorthandoff
401 \fi

```

Set the classes and interactions between classes.

```

402 \XeTeXinterchartokenstate=1
403 \XeTeXcharclass '\: = \FB@punctthick
404 \XeTeXinterchartoks \z@ \FB@punctthick = {%
405 \ifhmode\FDP@colonspace\fi}%
406 \XeTeXinterchartoks \FB@guilf \FB@punctthick = {%
407 \FDP@colonspace}%
408 \XeTeXinterchartoks 255 \FB@punctthick = {%
409 \ifhmode\unskip\penalty\M\FBcolonspace\fi}%
410 \@for\FB@char:={'\;,\;\,'!\,'?}\do
411 {\XeTeXcharclass\FB@char=\FB@punctthin}%
412 \XeTeXinterchartoks \z@ \FB@punctthin = {%
413 \ifhmode\FDP@thinspace\fi}%
414 \XeTeXinterchartoks \FB@guilf \FB@punctthin = {%
415 \FDP@thinspace}%
416 \XeTeXinterchartoks 255 \FB@punctthin = {%
417 \ifhmode\unskip\penalty\M\FBthinspace\fi}%
418 \XeTeXinterchartoks \FB@guilo \z@ = {%
419 \ifFBAutoSpaceGuill\FBguill@spacing\fi}%
420 \XeTeXinterchartoks \FB@guilo 255 = {%
421 \ifFBAutoSpaceGuill\FBguill@spacing\ignorespaces\fi}%

```

```

422 \XeTeXinterchartoks \z@ \FB@guilf = {%
423     \ifFBAutoSpaceGuill\FBguill@spacing\fi}%
424 \XeTeXinterchartoks \FB@punctthin \FB@guilf = {%
425     \ifFBAutoSpaceGuill\FBguill@spacing\fi}%
426 \XeTeXinterchartoks 255 \FB@guilf = {%
427     \ifFBAutoSpaceGuill\unskip\FBguill@spacing\fi}%

```

```

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

```

```

430 \@for\FB@char:={'\{, '\,, '\., '\-, '\), '\}, '\%, %
431 "22, "27, "60, "2019}\do
432 {\XeTeXcharclass\FB@char=\z@}%

```

```

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

```

```

445 \newcommand*{\FB@xetex@punct@nonfrench}{%
446   \ifx\FB@charclassesORI\empty
447   \else
448     \@for\FB@char:={\':,\';,\!','\?', "AB,"BB,%
449                     '\(,\'[, '\{,\',,\',,\'-,\') ,'\],'\},%
450                     '\%, "22,"27,"60,"2019,"A0,"202F}\do
451     {\expandafter\FB@parse\FB@charclassesORI\endparse
452      \XeTeXcharclass\FB@char=\FB@cclass}%
453   \def\FB@charclassesORI{}%
454   \XeTeXinterchartokenstate=\FB@interchartokenstateORI
455   \let\shorthandon\shorthandonORI
456   \let\shorthandoff\shorthandoffORI
457   \fi
458 }
459 \FB@addto{extras}{\FB@xetex@punct@french}
460 \FB@addto{noextras}{\FB@xetex@punct@nonfrench}

```


End of specific code for punctuation with modern XeTeX engines.

```
461 \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.

```
462 \ifFB@active@punct
463   \initiate@active@char{:}%
464   \initiate@active@char{;}%
465   \initiate@active@char{!}%
466   \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.

```
467   \declare@shorthand{french}{;}{;%
468     \ifhmode
469       \ifdim\lastskip>\z@
470         \unskip\penalty\M\FBthinspace
471       \else
472         \FDP@thinspace
473       \fi
474     \fi
```

Now we can insert a ; character.

```
475     \string;}
```

The next three definitions are very similar.

```
476   \declare@shorthand{french}{!}{;%
477     \ifhmode
478       \ifdim\lastskip>\z@
479         \unskip\penalty\M\FBthinspace
480       \else
481         \FDP@thinspace
482       \fi
483     \fi
484     \string!}
485   \declare@shorthand{french}{?}{;%
486     \ifhmode
487       \ifdim\lastskip>\z@
488         \unskip\penalty\M\FBthinspace
489       \else
490         \FDP@thinspace
491       \fi
492     \fi
493     \string?}
494   \declare@shorthand{french}{:}{;%
```

```

495     \ifhmode
496         \ifdim\lastskip>\z@
497             \unskip\penalty\@M\FBcolonspace
498         \else
499             \FDP@colonspace
500         \fi
501     \fi
502     \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.

```

503 \declare@shorthand{system}{:}{\string:}
504 \declare@shorthand{system}{!}{\string!}
505 \declare@shorthand{system}{?}{\string?}
506 \declare@shorthand{system}{;}{\string;}
507 %}

```

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

```

508 \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.

```

509 \bbl@activate{:}\bbl@activate{;}%
510 \bbl@activate{!}\bbl@activate{?}%
511 }
512 \FB@addto{noextras}{%
513     \bbl@deactivate{:}\bbl@deactivate{;}%
514     \bbl@deactivate{!}\bbl@deactivate{?}%
515 }
516 \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.

```

517 \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).

```

518 \def\autospace@beforeFDP{%
519     \ifFB@luatex@punct\FB@addDPspace=1 \fi

```

```

520         \def\FDP@thinspace{\penalty\@M\FBthinspace}%
521         \def\FDP@colonspace{\penalty\@M\FBcolonspace}}
522 \def\noautospace@beforeFDP{%
523     \ifFB@luatex@punct\FB@addDPSpace=0 \fi
524     \let\FDP@thinspace\@empty
525     \let\FDP@colonspace\@empty}
526 \ifLaTeXe
527     \def\AutoSpaceBeforeFDP{\autospace@beforeFDP
528         \FBAutoSpacePunctuationtrue}
529     \def\NoAutoSpaceBeforeFDP{\noautospace@beforeFDP
530         \FBAutoSpacePunctuationfalse}
531     \AtEndOfPackage{\AutoSpaceBeforeFDP}
532 \else
533     \let\AutoSpaceBeforeFDP\autospace@beforeFDP
534     \let\NoAutoSpaceBeforeFDP\noautospace@beforeFDP
535     \AutoSpaceBeforeFDP
536 \fi

```

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

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

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

```

537 \ifLaTeXe
538     %\let\ttfamilyORI\ttfamily
539     %\let\rmfamilyORI\rmfamily
540     %\let\sffamilyORI\sffamily
541     \DeclareRobustCommand\ttfamilyFB{%
542         \FBAutoSpaceGuillfalse
543         \ifFB@luatex@punct\FB@addGUILspace=0 \fi
544         \noautospace@beforeFDP\ttfamilyORI}%
545     \DeclareRobustCommand\rmfamilyFB{%
546         \FBAutoSpaceGuilltrue
547         \ifFB@luatex@punct\FB@addGUILspace=1 \fi
548         \ifFBAutoSpacePunctuation
549             \autospace@beforeFDP
550         \else
551             \noautospace@beforeFDP
552         \fi
553         \rmfamilyORI}%
554     \DeclareRobustCommand\sffamilyFB{%
555         \FBAutoSpaceGuilltrue
556         \ifFB@luatex@punct\FB@addGUILspace=1 \fi
557         \ifFBAutoSpacePunctuation

```

```

558         \autospace@beforeFDP
559     \else
560         \noautospace@beforeFDP
561     \fi
562     \sffamilyORI}%
563 \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.

```

564 \newcommand*{\NoAutoSpacing}{\FBAutoSpaceGuillfalse
565   \ifFB@active@punct\shorthandoff{;:!?}\fi
566   \ifFB@xetex@punct\XeTeXinterchartokenstate=0 \fi
567   \ifFB@luatex@punct\FB@addDPspace=0 \FB@addGUILspace=0 \fi
568 }

```

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 multilingual texts is to use the package `csquotes` and its command `\enquote`. 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`.

```

569 \newcommand*{\og}{\@empty}
570 \newcommand*{\fg}{\@empty}

```

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

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

```

571 \ifLaTeXe
572 \else
573   \ifFBunicode
574     \def\guillemotleft{{\char"00AB}}
575     \def\guillemotright{{\char"00BB}}
576     \def\textquotedblleft{{\char"201C}}
577     \def\textquotedblright{{\char"201D}}
578   \else
579     \def\guillemotleft{\leavevmode\raise0.25ex
580       \hbox{$\scriptscriptstyle\ll$}}
581     \def\guillemotright{\raise0.25ex
582       \hbox{$\scriptscriptstyle\gg$}}
583     \def\textquotedblleft{' '}

```

```

584 \def\textquotedblright{''}
585 \fi
586 \let\xspace\relax
587 \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. `\FBguill@spacing` 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 `\FBguill@spacing` for the `lmr10` font, its dimensions will be scaled by `frenchb.lua` for the current font.

```

588 \newskip\FBguillskip
589 \FBguillskip=2.664pt plus 0.500pt minus 0.888pt \relax
590 \newcommand*{\FBguill@spacing}{\penalty\@M\hskip.8\fontdimen2\font
591                                     plus.3\fontdimen3\font
592                                     minus.8\fontdimen4\font}

```

`\FBguill@spacing` is not used with LuaTeX.

```

593 \ifFB@luatex@punct
594 \DeclareRobustCommand*{\FB@og}{\leavevmode
595     \bgroup\FB@addGUllspace=1 \guillemotleft\egroup}
596 \DeclareRobustCommand*{\FB@fg}{\ifdim\lastskip>\z@\unskip\fi
597     \bgroup\FB@addGUllspace=1 \guillemotright\egroup\xspace}
598 \fi

```

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

```

599 \ifFB@xetex@punct
600 \DeclareRobustCommand*{\FB@og}{\leavevmode
601     \bgroup\FBAutoSpaceGuillfalse\guillemotleft\egroup
602     \FBguill@spacing}
603 \DeclareRobustCommand*{\FB@fg}{\ifdim\lastskip>\z@\unskip\fi
604     \FBguill@spacing
605     \bgroup\FBAutoSpaceGuillfalse\guillemotright\egroup\xspace}
606 \fi
607 \ifFB@active@punct
608 \DeclareRobustCommand*{\FB@og}{\leavevmode
609     \guillemotleft
610     \FBguill@spacing}
611 \DeclareRobustCommand*{\FB@fg}{\ifdim\lastskip>\z@\unskip\fi
612     \FBguill@spacing
613     \guillemotright\xspace}
614 \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.

```

615 \ifLaTeXe
616   \def\bbf@frenchguillemets{\renewcommand*{\og}{\FB@og}%
617                               \renewcommand*{\fg}{\FB@fg}}
618   \renewcommand*{\og}{\textquotedblleft}
619   \renewcommand*{\fg}{\ifdim\lastskip>\z@ \unskip\fi \textquotedblright}
620 \else
621   \def\bbf@frenchguillemets{\let\og\FB@og
622                               \let\fg\FB@fg}
623   \def\og{\textquotedblleft}
624   \def\fg{\ifdim\lastskip>\z@ \unskip\fi \textquotedblright}
625 \fi
626 \FB@addto{extras}{\babel@save\og \babel@save\fg \bbf@frenchguillemets}

```

2.4 Date in French

\datefrench The macro `\datefrench` redefines the command `\today` to produce French dates. This new implementation requires babel 3.9 or newer.

```

627 \ifLaTeXe
628   \def\BabelLanguages{french,acadian}
629   \StartBabelCommands*{\BabelLanguages}{date}
630     [unicode, fontenc=EU1 EU2, charset=utf8]
631     \SetString\monthiiname{février}
632     \SetString\monthviiiname{août}
633     \SetString\monthxiiname{décembre}
634   \StartBabelCommands*{\BabelLanguages}{date}
635     \SetStringLoop{month#1name}{%
636       janvier,f\'evrier,mars,avril,mai,juin,juillet,%
637       ao\^ut,septembre,octobre,novembre,d\'ecembre}
638     \SetString\today{{\number\day}\ifnum1=\day {\ier}\fi \space
639       \csname month\romannumeral\month name\endcsname \space
640       \number\year
641     }
642   \EndBabelCommands
643 \else
644   \ifFBunicode
645     \@namedef{date\CurrentOption}{%
646       \def\today{{\number\day}\ifnum1=\day {\ier}\fi \space
647         \ifcase\month
648           \or janvier\or février\or mars\or avril\or mai\or
649           juin\or juillet\or août\or septembre\or
650           octobre\or novembre\or décembre\fi
651         \space \number\year}}
652   \else
653     \@namedef{date\CurrentOption}{%
654       \def\today{{\number\day}\ifnum1=\day {\ier}\fi \space
655         \ifcase\month
656           \or janvier\or f\'evrier\or mars\or avril\or mai\or

```

```

657         juin\or juillet\or ao^ut\or septembre\or
658         octobre\or novembre\or d\'ecembre\fi
659     \space \number\year}}
660 \fi
661 \fi

```

2.5 Extra utilities

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

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

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

```

662 \newif\ifFB@poorman
663 \newdimen\FB@Mht
664 \ifLaTeXe
665   \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{}`.

```

666   \newcommand*{\FBsupR}{-0.12}
667   \newcommand*{\FBsupS}{0.65}
668   \newcommand*{\FB@lc}[1]{\MakeLowercase{#1}}
669   \DeclareRobustCommand*{\FB@up@fake}[1]{%
670     \settoheight{\FB@Mht}{M}%
671     \addtolength{\FB@Mht}{\FBsupR \FB@Mht}%
672     \addtolength{\FB@Mht}{-\FBsupS ex}%
673     \raisebox{\FB@Mht}{\scalefont{\FBsupS}{\FB@lc{#1}}}%
674   }

```

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.

```

675 \def\FB@split#1#2#3#4\@nil{\def\FB@firstthree{#1#2#3}%
676                               \def\FB@suffix{#4}}
677 \def\FB@x{x}
678 \def\FB@j{j}
679 \DeclareRobustCommand*\FB@up}[1]{%
680   \bgroup \FB@poormantrue
681   \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.

```

682   \edef\reserved@a{\lowercase{%
683     \noexpand\IfFileExists{\f@encoding\FB@firstthree -sup.fd}}}%
684   \reserved@a
685   {\ifx\FB@suffix\FB@x \FB@poormanfalse\fi
686    \ifx\FB@suffix\FB@j \FB@poormanfalse\fi
687    \ifFB@poorman \FB@up@fake{#1}%
688    \else        \FB@up@real{#1}%
689    \fi}%
690   {\FB@up@fake{#1}}%
691   \egroup}

```

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

```

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

```

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

```

694 \DeclareRobustCommand*\fup}[1]{%
695   \ifx\realsuperscript\@undefined
696     \FB@up{#1}%
697   \else
698     \bgroup\let\fakesuperscript\FB@up@fake
699     \realsuperscript{\FB@lc{#1}}\egroup
700   \fi}

```

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

```

701 \providecommand*\up{\relax}

```

Poor man’s definition of \up for Plain.

```

702 \else
703 \providecommand*\up}[1]{\leavevmode\raiselex\hbox{\sevenrm #1}}

```


704 \fi

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

```
\ier 705 \def\ieme{\up{\lowercase{e}}\xspace}
\iere 706 \def\iemes{\up{\lowercase{es}}\xspace}
\iemes 707 \def\ier{\up{\lowercase{er}}\xspace}
\iers 708 \def\iers{\up{\lowercase{ers}}\xspace}
\ieres 709 \def\iere{\up{\lowercase{re}}\xspace}
710 \def\ieres{\up{\lowercase{res}}\xspace}
```

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

```
\no 711 \newcommand*{\FrenchEnumerate}[1]{%
\nos 712 #1\up{\lowercase{o}}\kern+.3em}
\nos 713 \newcommand*{\FrenchPopularEnumerate}[1]{%
\primo 714 #1\up{\lowercase{o}}\kern+.3em}
\frprimo) Typing \primo should result in '1°',
```

```
715 \def\primo{\FrenchEnumerate1}
716 \def\secundo{\FrenchEnumerate2}
717 \def\tertio{\FrenchEnumerate3}
718 \def\quarto{\FrenchEnumerate4}
```

while typing \frprimo) gives '1°'.

```
719 \def\frprimo{\FrenchPopularEnumerate1}
720 \def\frsecundo{\FrenchPopularEnumerate2}
721 \def\frtertio{\FrenchPopularEnumerate3}
722 \def\frquarto{\FrenchPopularEnumerate4}
```

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

```
723 \DeclareRobustCommand*\No{\N\up{\lowercase{o}}\kern+.2em}
724 \DeclareRobustCommand*\no{\n\up{\lowercase{o}}\kern+.2em}
725 \DeclareRobustCommand*\Nos{\N\up{\lowercase{os}}\kern+.2em}
726 \DeclareRobustCommand*\nos{\n\up{\lowercase{os}}\kern+.2em}
```

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

```
727 \DeclareRobustCommand*\bsc[1]{\leavevmode\begin{group}\kern0pt
728 \scshape #1\end{group}}
729 \ifLaTeX\else\let\scshape\relax\fi
```

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

```
730 \ifFBUnicode
731 \newcommand*\at{{\char"0040}}
```

```

732 \newcommand*\circonflexe{{\char"005E}}
733 \newcommand*\tild{{\char"007E}}
734 \newcommand*\boi{{\textbackslash}}
735 \newcommand*\degree{{\char"00B0}}
736 \else
737 \ifLaTeXe
738 \DeclareTextSymbol{\at}{T1}{64}
739 \DeclareTextSymbol{\circonflexe}{T1}{94}
740 \DeclareTextSymbol{\tild}{T1}{126}
741 \DeclareTextSymbolDefault{\at}{T1}
742 \DeclareTextSymbolDefault{\circonflexe}{T1}
743 \DeclareTextSymbolDefault{\tild}{T1}
744 \DeclareRobustCommand*\boi{{\textbackslash}}
745 \DeclareRobustCommand*\degree{\r{}}
746 \else
747 \def\T@one{T1}
748 \ifx\fontencoding\T@one
749 \newcommand*\degree{{\char6}}
750 \else
751 \newcommand*\degree{{\char23}}
752 \fi
753 \newcommand*\at{{\char64}}
754 \newcommand*\circonflexe{{\char94}}
755 \newcommand*\tild{{\char126}}
756 \newcommand*\boi{{\backslash}}
757 \fi
758 \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 0.3em, 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.

```

759 \ifLaTeXe
760 \newcommand*\degrees{\degree}
761 \ifFBunicode
762 \DeclareRobustCommand*\degrees{\degree}
763 \else
764 \def\Warning@degree@TSone{%
765 \PackageWarning{frenchb.ldf}{%
766 Degrees would look better in TS1-encoding:%
767 \MessageBreak add \protect
768 \usepackage{textcomp} to the preamble.%
769 \MessageBreak Degrees used}}
770 \AtBeginDocument{\ifx\DeclareEncodingSubset\@undefined
771 \DeclareRobustCommand*\degrees{%
772 \leavevmode\hbox to 0.3em{\hss\degree\hss}%

```

```

773             \Warning@degree@TSone
774             \global\let\Warning@degree@TSone\relax}%
775         \else
776             \DeclareRobustCommand*{\degres}{%
777                 \hbox{\UseTextSymbol{TS1}{\textdegree}}}%
778         \fi
779     }
780 \fi
781 \else
782     \newcommand*{\degres}{%
783         \leavevmode\hbox to 0.3em{\hss\degre\hss}}
784 \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 `{,}`. **\DecimalMathComma** makes the comma be an ordinary character (of type `\mathord`) in French *only* (no space added); **\StandardMathComma** switches back to the standard behaviour of the comma.

```

785 \newcount\std@mcc
786 \newcount\dec@mcc
787 \std@mcc=\mathcode'\,
788 \dec@mcc=\std@mcc
789 \@tempcnta=\std@mcc
790 \divide\@tempcnta by "1000
791 \multiply\@tempcnta by "1000
792 \advance\dec@mcc by -\@tempcnta
793 \newcommand*{\DecimalMathComma}{\iflanguage{french}%
794     {\mathcode'\,=\dec@mcc}}}%
795     \FB@addto{extras}{\mathcode'\,=\dec@mcc}%
796 }
797 \newcommand*{\StandardMathComma}{\mathcode'\,=\std@mcc
798     \FB@addto{extras}{\mathcode'\,=\std@mcc}%
799 }
800 \FB@addto{noextras}{\mathcode'\,=\std@mcc}

```

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

```

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

```

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

```

803 \let\FBstop@here\relax
804 \def\FBclean@on@exit{\let\ifLaTeXe\undefined
805                     \let\LaTeXetrue\undefined
806                     \let\LaTeXefalse\undefined}
807 \ifx\magnification\@undefined
808 \else
809   \def\FBstop@here{\ifFB@luatex@punct
810                   \activate@luatexpunct
811                   \fi
812                   \FBclean@on@exit
813                   \ldf@quit\CurrentOption\endinput}
814 \fi
815 \FBstop@here

```

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

```

816 \renewcommand*{\nombre}[1]{\Warning@nombre\numprint{#1}}
817 \newcommand*{\Warning@nombre}{%
818   \ifdefined\numprint
819   \else
820     \PackageWarning{frenchb.ldf}{%
821       \protect\nombre\space now relies on package numprint.sty,%
822       \MessageBreak add \protect
823       \usepackage[autolanguage]{numprint}\MessageBreak
824       to your preamble *after* loading babel,\MessageBreak
825       see file numprint.pdf for more options.\MessageBreak
826       \protect\nombre\space called}%
827     \global\let\Warning@nombre\relax
828   \fi
829 }

```

2.7 Caption names

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

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

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

```

830 \StartBabelCommands*{\BabelLanguages}{captions}
831   [unicode, fontenc=EU1 EU2, charset=utf8]
832   \SetString{\refname}{Références}
833   \SetString{\abstractname}{Résumé}
834   \SetString{\prefacename}{Préface}
835   \SetString{\contentsname}{Table des matières}
836   \SetString{\ccname}{Copie à }
837   \SetString{\proofname}{Démonstration}

```

```

838 \SetStringLoop{ordinal#1}{%
839     Première,Deuxième,Troisième,Quatrième,Cinquième,%
840     Sixième,Septième,Huitième,Neuvième,Dixième,Onzième,%
841     Douzième,Treizième,Quatorzième,Quinzième,Seizième,%
842     Dix-septième,Dix-huitième,Dix-neuvième,Vingtième}
843 \StartBabelCommands*{\BabelLanguages}{captions}
844 \SetString{\refname}{R\ref\erences}
845 \SetString{\abstractname}{R\esum\e}
846 \SetString{\bibname}{Bibliographie}
847 \SetString{\prefacename}{Pr\eface}
848 \SetString{\chaptername}{Chapitre}
849 \SetString{\appendixname}{Annexe}
850 \SetString{\contentsname}{Table des mati\eres}
851 \SetString{\listfigurename}{Table des figures}
852 \SetString{\listtablename}{Liste des tableaux}
853 \SetString{\indexname}{Index}
854 \SetString{\figurename}{{\scshape Figure}}
855 \SetString{\tablename}{{\scshape Table}}
856 \SetString{\pagename}{page}
857 \SetString{\seename}{voir}
858 \SetString{\alsoname}{voir aussi}
859 \SetString{\enclname}{P.~J. }
860 \SetString{\ccname}{Copie \a }
861 \SetString{\headtoname}{}
862 \SetString{\proofname}{D\emonstration}
863 \SetString{\glossaryname}{Glossaire}
    "Première partie" instead of "Part I".
864 \SetStringLoop{ordinal#1}{%
865     Premi\ere,Deuxi\eme,Troisi\eme,Quatri\eme,Cinqui\eme,%
866     Sixi\eme,Septi\eme,Huiti\eme,Neuvi\eme,Dixi\eme,Onzi\eme,%
867     Douzi\eme,Treizi\eme,Quatorzi\eme,Quinzi\eme,Seizi\eme,%
868     Dix-septi\eme,Dix-huiti\eme,Dix-neuvi\eme,Vingti\eme}
869 \AfterBabelCommands{%
870     \DeclareRobustCommand*{\FB@emptypart}{\def\thepart{}}%
871     \DeclareRobustCommand*{\FB@Rpart}{\def\thepart{\Roman{part}}}%
872 }
873 \SetString{\partname}{%
874     \csname ordinal\romannumeral\value{part}\endcsname\space
875     partie\FB@emptypart}
876 \EndBabelCommands

```

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

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

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

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

```

877 \bgroup
878   \catcode'::=12 \catcode'>:=12 \relax
879   \long\gdef\STD@makecaption#1#2{%
880     \vskip\abovcaptionskip
881     \sbox\@tempboxa{#1: #2}%
882     \ifdim \wd\@tempboxa >\hsize
883       #1: #2\par
884     \else
885       \global \@minipagefalse
886       \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
887     \fi
888     \vskip\belowcaptionskip}
889 \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.

```

890 \newif\if@FBwarning@capsep
891 \@FBwarning@capseptrue
892 \newcommand{\FBWarning}[2]{\PackageWarning{#1}{#2}}
893 \newcommand*{\CaptionSeparator}{\space\textendash\space}
894 \def\FBCaption@Separator{:\space}
895 \long\def\FB@makecaption#1#2{%
896   \vskip\abovcaptionskip
897   \sbox\@tempboxa{#1\FBCaption@Separator #2}%
898   \ifdim \wd\@tempboxa >\hsize
899     #1\FBCaption@Separator #2\par
900   \else
901     \global \@minipagefalse
902     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
903   \fi

```

904 \vskip\belowcaptionskip}

Disable the standard warning with AMS and SMF classes.

```
905 \@ifclassloaded{amsart}{\@FBwarning@capsepfalse}{}
906 \@ifclassloaded{amsbook}{\@FBwarning@capsepfalse}{}
907 \@ifclassloaded{amstex}{\@FBwarning@capsepfalse}{}
908 \@ifclassloaded{amslatex}{\@FBwarning@capsepfalse}{}
909 \@ifclassloaded{amproc}{\@FBwarning@capsepfalse}{}
910 \@ifclassloaded{smfart}{\@FBwarning@capsepfalse}{}
911 \@ifclassloaded{smfbook}{\@FBwarning@capsepfalse}{}

```

Disable the standard warning unless high punctuation is active.

```
912 \ifFB@active@punct\else\@FBwarning@capsepfalse\fi

```

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

```
913 \newif\ifFB@koma
914 \@ifclassloaded{memoir}{\@FBwarning@capsepfalse}{}
915 \@ifclassloaded{scrartcl}{\@FBwarning@capsepfalse\FB@komatrue}{}
916 \@ifclassloaded{scrbook}{\@FBwarning@capsepfalse\FB@komatrue}{}
917 \@ifclassloaded{scrreprt}{\@FBwarning@capsepfalse\FB@komatrue}{}
918 \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.

```
919 \@ifpackageloaded{caption}
920   {\FBWarning{frenchb.ldf}%
921     {Please load the "caption" package\MessageBreak
922       AFTER babel/frenchb; reported}%
923   \@FBwarning@capsepfalse}%
924   {}

```

Same for package floatrow.

```
925 \@ifpackageloaded{floatrow}
926   {\FBWarning{frenchb.ldf}%
927     {Please load the "floatrow" package\MessageBreak
928       AFTER babel/frenchb; reported}%
929   \@FBwarning@capsepfalse}%
930   {}

```

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’).

```
931 \AtBeginDocument{%
932   \ifx\@makecaption\STD@makecaption
933     \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.

```

934 \ifFBoldFigTabCaptions
935 \else
936 \def\FBCaption@Separator{\iflanguage{french}{ : }{: }}%
937 \fi
938 \ifFBCustomiseFigTabCaptions
939 \ifx\bbl@main@language\FB@french
940 \def\FBCaption@Separator{\CaptionSeparator}%
941 \fi
942 \fi
943 \@FBwarning@capsepfalse
944 \fi
945 \if@FBwarning@capsep
946 \FBWarning{frenchb.ldb}%
947 {Figures' and tables' captions might look like\MessageBreak
948 'Figure 1:' which is wrong in French.\MessageBreak
949 Check your class or packages to change this;\MessageBreak
950 reported}%
951 \fi
952 \let\FB@makecaption\relax
953 \let\STD@makecaption\relax
954 }

```

2.8 Dots...

\FBtextellipsis $\text{\LaTeX}_{2\epsilon}$'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 $\text{\LaTeX}_{2\epsilon}$ only).

The `\if` construction in the $\text{\LaTeX}_{2\epsilon}$ 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.

```

955 \ifFBunicode
956 \let\FBtextellipsis\textellipsis
957 \else
958 \DeclareTextSymbol{\FBtextellipsis}{LY1}{133}
959 \DeclareTextCommandDefault{\FBtextellipsis}{%
960 \kern\fontdimen3\font.\kern\fontdimen3\font.\xspace}
961 \fi

```

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

```

962 \newcommand*{\Tdots@}{\@xp\textellipsis}
963 \newcommand*{\Mdots@}{\@xp\mdots@}

```



```

964 \AtBeginDocument{\DeclareRobustCommand*\dots{\relax
965             \csname\ifmmode M\else T\fi dots@endcsname}%
966             \ifdefined\@xp\else\let\@xp\relax\fi
967             \ifdefined\mdots\else\let\Mdots@mathellipsis\fi
968             }
969 \def\bbl@frenchdots{\babel@save\Tdots@ \let\Tdots@\FBtextellipsis}
970 \FB@addto{extras}{\bbl@frenchdots}

```

2.9 Setup options: keyval stuff

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

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

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

```

971 \newcommand*\frenchbsetup[1]{%
972   \setkeys{FB}{#1}%
973 }%
974 \@onlypreamble\frenchbsetup

```

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

```

975 \newif\ifFBStandardLayout           \FBStandardLayouttrue
976 \newif\ifFBGlobalLayoutFrench       \FBGlobalLayoutFrenchtrue
977 \newif\ifFBReduceListSpacing        \FBReduceListSpacingfalse
978 \newif\ifFBListOldLayout            \FBListOldLayoutfalse
979 \newif\ifFBCompactItemize           \FBCompactItemizefalse
980 \newif\ifFBStandardItemizeEnv        \FBStandardItemizeEnvtrue
981 \newif\ifFBStandardEnumerateEnv      \FBStandardEnumerateEnvtrue
982 \newif\ifFBStandardItemLabels        \FBStandardItemLabelstrue
983 \newif\ifFBStandardLists            \FBStandardListstrue
984 \newif\ifFBIndentFirst              \FBIndentFirstfalse
985 \newif\ifFBFrenchFootnotes          \FBFrenchFootnotesfalse
986 \newif\ifFBAutoSpaceFootnotes       \FBAutoSpaceFootnotesfalse
987 \newif\ifFBOriginalTypewriter       \FBOriginalTypewriterfalse
988 \newif\ifFBThinColonSpace           \FBThinColonSpacefalse
989 \newif\ifFBThinSpaceInFrenchNumbers \FBThinSpaceInFrenchNumbersfalse
990 \newif\ifFBFrenchSuperscripts       \FBFrenchSuperscriptstrue
991 \newif\ifFBLowercaseSuperscripts    \FBLowercaseSuperscriptstrue

```

```

992 \newif\ifFBPartNameFull          \FBPartNameFulltrue
993 \newif\ifFBCustomiseFigTabCaptions \FBCustomiseFigTabCaptionsfalse
994 \newif\ifFBOldFigTabCaptions      \FBOldFigTabCaptionsfalse
995 \newif\ifFBSuppressWarning         \FBSuppressWarningfalse
996 \newif\ifFBShowOptions             \FBShowOptionsfalse

```

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

```

997 \edef\FB@french{\CurrentOption}
998 \AtEndOfPackage{%
999   \ifx\bbl@main@language\FB@french
1000     \FBGlobalLayoutFrenchtrue
1001     \FBReduceListSpacingtrue
1002     \FBStandardItemizeEnvfalse
1003     \FBStandardEnumerateEnvfalse
1004     \FBStandardItemLabelsfalse
1005     \FBIndentFirsttrue
1006     \FBFrenchFootnotesttrue
1007     \FBAutoSpaceFootnotesttrue
1008     \FBCustomiseFigTabCaptionstrue
1009   \else
1010     \FBGlobalLayoutFrenchfalse
1011   \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.

```

1012 \RequirePackage{keyval}%
1013 \define@key{FB}{StandardLayout}[true]%
1014   {\csname FBStandardLayout#1\endcsname
1015    \ifFBStandardLayout
1016      \FBReduceListSpacingfalse
1017      \FBStandardItemizeEnvtrue
1018      \FBStandardItemLabelstrue
1019      \FBStandardEnumerateEnvtrue
1020      \FBIndentFirstfalse
1021      \FBFrenchFootnotesfalse
1022      \FBAutoSpaceFootnotesfalse
1023      \FBGlobalLayoutFrenchfalse
1024    \else
1025      \FBReduceListSpacingtrue
1026      \FBStandardItemizeEnvfalse
1027      \FBStandardItemLabelsfalse
1028      \FBStandardEnumerateEnvfalse
1029      \FBIndentFirsttrue
1030      \FBFrenchFootnotesttrue

```

```

1031          \FBAutoSpaceFootnotesttrue
1032          \fi}%
1033 \define@key{FB}{GlobalLayoutFrench}[true]%
1034          {\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.

```

1035          \ifFBGlobalLayoutFrench
1036          \ifx\bbbl@main@language\FB@french
1037          \else
1038          \PackageWarning{frenchb.ldf}%
1039              {Option 'GlobalLayoutFrench' skipped:%
1040              \MessageBreak French is *not*
1041              babel's last option.\MessageBreak}%
1042          \fi
1043          \fi}%
1044 \define@key{FB}{ReduceListSpacing}[true]%
1045          {\csname FBReduceListSpacing#1\endcsname}%
1046 \define@key{FB}{ListOldLayout}[true]%
1047          {\csname FBListOldLayout#1\endcsname
1048          \ifFBListOldLayout
1049          \FBStandardEnumerateEnvtrue
1050          \renewcommand*{\FrenchLabelItem}{\textendash}%
1051          \fi}%
1052 \define@key{FB}{CompactItemize}[true]%
1053          {\csname FBCompactItemize#1\endcsname
1054          \ifFBCompactItemize
1055          \FBStandardItemizeEnvfalse
1056          \FBStandardEnumerateEnvfalse
1057          \else
1058          \FBStandardItemizeEnvtrue
1059          \FBStandardEnumerateEnvtrue
1060          \fi}%
1061 \define@key{FB}{StandardItemizeEnv}[true]%
1062          {\csname FBStandardItemizeEnv#1\endcsname}%
1063 \define@key{FB}{StandardEnumerateEnv}[true]%
1064          {\csname FBStandardEnumerateEnv#1\endcsname}%
1065 \define@key{FB}{StandardItemLabels}[true]%
1066          {\csname FBStandardItemLabels#1\endcsname}%
1067 \define@key{FB}{ItemLabels}{%
1068     \renewcommand*{\FrenchLabelItem}{#1}}%
1069 \define@key{FB}{ItemLabeli}{%
1070     \renewcommand*{\Frlabelitemi}{#1}}%
1071 \define@key{FB}{ItemLabelii}{%
1072     \renewcommand*{\Frlabelitemii}{#1}}%
1073 \define@key{FB}{ItemLabeliii}{%
1074     \renewcommand*{\Frlabelitemiii}{#1}}%
1075 \define@key{FB}{ItemLabeliv}{%
1076     \renewcommand*{\Frlabelitemiv}{#1}}%
1077 \define@key{FB}{StandardLists}[true]%

```

```

1078         {\csname FBStandardLists#1\endcsname
1079         \ifFBStandardLists
1080             \FBReduceListSpacingfalse
1081             \FBCompactItemizefalse
1082             \FBStandardItemizeEnvtrue
1083             \FBStandardEnumerateEnvtrue
1084             \FBStandardItemLabelstrue
1085         \else
1086             \FBReduceListSpacingtrue
1087             \FBCompactItemizetrue
1088             \FBStandardItemizeEnvfalse
1089             \FBStandardEnumerateEnvfalse
1090             \FBStandardItemLabelsfalse
1091         \fi}%
1092 \define@key{FB}{IndentFirst}[true]%
1093         {\csname FBIndentFirst#1\endcsname}%
1094 \define@key{FB}{FrenchFootnotes}[true]%
1095         {\csname FBFrenchFootnotes#1\endcsname}%
1096 \define@key{FB}{AutoSpaceFootnotes}[true]%
1097         {\csname FBAutoSpaceFootnotes#1\endcsname}%
1098 \define@key{FB}{AutoSpacePunctuation}[true]%
1099         {\csname FBAutoSpacePunctuation#1\endcsname}%
1100 \define@key{FB}{OriginalTypewriter}[true]%
1101         {\csname FBOriginalTypewriter#1\endcsname}%
1102 \define@key{FB}{ThinColonSpace}[true]%
1103         {\csname FBThinColonSpace#1\endcsname}%
1104 \define@key{FB}{ThinSpaceInFrenchNumbers}[true]%
1105         {\csname FBThinSpaceInFrenchNumbers#1\endcsname}%
1106 \define@key{FB}{FrenchSuperscripts}[true]%
1107         {\csname FBFrenchSuperscripts#1\endcsname}%
1108 \define@key{FB}{LowercaseSuperscripts}[true]%
1109         {\csname FBLowercaseSuperscripts#1\endcsname}%
1110 \define@key{FB}{PartNameFull}[true]%
1111         {\csname FBPartNameFull#1\endcsname
1112         \ifFBPartNameFull
1113         \else
1114             \FB@addto{captions}{%
1115             \def\partname{Partie\protect\FB@Rpart}}}%
1116         \fi}%
1117 \define@key{FB}{CustomiseFigTabCaptions}[true]%
1118         {\csname FBCustomiseFigTabCaptions#1\endcsname}%
1119 \define@key{FB}{OldFigTabCaptions}[true]%
1120         {\csname FBOldFigTabCaptions#1\endcsname
        \CurrentOption no longer defined. It's value has been saved in \FB@CurOpt while
        reading frenchb. ldf.
1121         \ifFBOldFigTabCaptions
1122             \FB@addto{extras}{\babel@save\FBCaption@Separator
1123             \def\FBCaption@Separator{\CaptionSeparator}}}%
1124         \fi}%
1125 \define@key{FB}{SuppressWarning}[true]%

```

```

1126             {\csname FBSuppressWarning#1\endcsname
1127             \ifFBSuppressWarning
1128               \renewcommand{\FBWarning}[2]{\relax}%
1129             \fi}%
1130 \define@key{FB}{ShowOptions}[true]%
1131             {\csname FBShowOptions#1\endcsname}%

```

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

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

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

```

1132 \define@key{FB}{og}{%
1133   \ifFB@active@punct
1134     \newcommand*{\FB@@og}{%
1135       \iflanguage{french}%
1136         {\ifFBAutoSpaceGuill\FB@og\ignorespaces
1137          \else\guillemotleft
1138          \fi}%
1139       {\guillemotleft}}%
1140   \fi
1141   \ifFBunicode

```

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

```

1142   \ifFB@luatex@punct
1143     \FB@addGUILspace=1 \relax
1144   \else

```

then modern XeTeX, this more tricky:

```

1145   \ifFB@xetex@punct

```

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

```

1146     \XeTeXcharclass"AB = \FB@guilo
1147     \XeTeXcharclass"A0 = \FB@guilnul
1148     \XeTeXcharclass"202F = \FB@guilnul
1149   \else

```

then an old LuaTeX or XeTeX in use, the following trick for defining the active quote character is borrowed from `inputenc.dtx`.

```

1150     \catcode'#1=\active

```

```

1151         \bgroup
1152         \uccode'\~'#1%
1153         \uppercase{%
1154         \egroup
1155         \def~%
1156         }{\FB@@og}%
1157     \fi
1158 \fi
1159 \else
    This is for conventional TeX engines:
1160     \AtBeginDocument{%
1161     \ifdefined\DeclareInputText
1162     \ifdefined\uc@dccl
        Package inputenc with utf8x encoding loaded, use \uc@dccl,
1163         \uc@dccl{171}{default}{\FB@@og}%
1164     \else
        if encoding is not utf8x, try utf8...
1165         \ifdefined\DeclareUnicodeCharacter
            utf8 loaded, use \DeclareUnicodeCharacter,
1166             \DeclareUnicodeCharacter{00AB}{\FB@@og}%
1167         \else
            if utf8 is not loaded either, we assume 8-bit character input encoding. Package
            MULEenc (from CJK) defines \mule@def to map characters to control sequences.
1168             \@tempcnta'#1\relax
1169             \ifdefined\mule@def
1170                 \mule@def{11}{\FB@@og}%
1171             \else
1172                 \DeclareInputText{\the\@tempcnta}{\FB@@og}%
1173             \fi
1174         \fi
1175     \fi
1176 \else
    Package inputenc not loaded, no way...
1177     \PackageWarning{frenchb.ldb}%
1178     {Option 'og' requires package inputenc.\MessageBreak}%
1179 \fi
1180 }%
1181 \fi
1182 }%

    Same code for the closing quote.
1183 \define@key{FB}{fg}{%
1184     \ifFB@active@punct
1185     \newcommand*{\FB@fg}{%
1186     \iflanguage{french}%
1187     {\ifFBAutoSpaceGuill\FB@fg
1188     \else\guillemotright
1189     \fi}%

```

```

1190         {\guillemotright}}}%
1191     \fi
1192     \ifFBunicode
1193         \ifFB@luatex@punct
1194             \FB@addGUILspace=1 \relax
1195         \else
1196             \ifFB@xetex@punct
1197                 \XeTeXcharclass"BB = \FB@guilf
1198                 \XeTeXcharclass"A0 = \FB@guilnul
1199                 \XeTeXcharclass"202F = \FB@guilnul
1200             \else
1201                 \catcode'#1=\active
1202                 \bgroup
1203                 \uccode'\~'#1%
1204                 \uppercase{%
1205                 \egroup
1206                 \def~%
1207                 }{{\FB@@fg}}}%
1208             \fi
1209         \fi
1210     \else
1211         \AtBeginDocument{%
1212             \ifdefined\DeclareInputText
1213                 \ifdefined\uc@dcl
1214                     \uc@dcl{187}{default}{\FB@@fg}%
1215                 \else
1216                     \ifdefined\DeclareUnicodeCharacter
1217                         \DeclareUnicodeCharacter{00BB}{\FB@@fg}%
1218                     \else
1219                         \@tempcnta'#1\relax
1220                         \ifdefined\mule@def
1221                             \mule@def{27}{\FB@@fg}%
1222                         \else
1223                             \DeclareInputText{\the\@tempcnta}{\FB@@fg}%
1224                         \fi
1225                     \fi
1226                 \fi
1227             \else
1228                 \PackageWarning{frenchb.ldb}%
1229                 {Option 'fg' requires package inputenc.\MessageBreak}%
1230             \fi
1231         }%
1232     \fi
1233 }%
1234 }

```

\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`.

```
1235 \newcommand*{\FBprocess@options}{%
```

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

```
1236 \@ifpackageloaded{enumitem}{%
1237   \ifFBStandardItemizeEnv
1238   \else
1239     \FBStandardItemizeEnvtrue
1240     \PackageInfo{frenchb.ldb}{%
1241       {Setting StandardItemizeEnv=true for\MessageBreak
1242         compatibility with enumitem package,\MessageBreak}%
1243     \fi
1244     \ifFBStandardEnumerateEnv
1245     \else
1246       \FBStandardEnumerateEnvtrue
1247       \PackageInfo{frenchb.ldb}{%
1248         {Setting StandardEnumerateEnv=true for\MessageBreak
1249           compatibility with enumitem package,\MessageBreak}%
1250     \fi}}%
1251 \@ifpackageloaded{paralist}{%
1252   \ifFBStandardItemizeEnv
1253   \else
1254     \FBStandardItemizeEnvtrue
1255     \PackageInfo{frenchb.ldb}{%
1256       {Setting StandardItemizeEnv=true for\MessageBreak
1257         compatibility with paralist package,\MessageBreak}%
1258     \fi
1259     \ifFBStandardEnumerateEnv
1260     \else
1261       \FBStandardEnumerateEnvtrue
1262       \PackageInfo{frenchb.ldb}{%
1263         {Setting StandardEnumerateEnv=true for\MessageBreak
1264           compatibility with paralist package,\MessageBreak}%
1265     \fi}}%
1266 \@ifpackageloaded{enumerate}{%
1267   \ifFBStandardEnumerateEnv
1268   \else
1269     \FBStandardEnumerateEnvtrue
1270     \PackageInfo{frenchb.ldb}{%
1271       {Setting StandardEnumerateEnv=true for\MessageBreak
1272         compatibility with enumerate package,\MessageBreak}%
1273   \fi}}%
```

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

```
1274 \def\FB@ufl{\update@frenchlists}
1275 \ifx\bbl@main@language\FB@french
1276   \update@frenchlists
1277 \fi
```


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

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

```
1278 \ifBFAutoSpacePunctuation
1279     \autospace@beforeFDP
1280 \else
1281     \noautospace@beforeFDP
1282 \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.

```
1283 \ifFBOriginalTypewriter
1284 \else
1285     \let\ttfamilyORI\ttfamily
1286     \let\rmfamilyORI\rmfamily
1287     \let\sffamilyORI\sffamily
1288     \let\ttfamily\ttfamilyFB
1289     \let\rmfamily\rmfamilyFB
1290     \let\sffamily\sffamilyFB
1291 \fi
```

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

```
1292 \ifFBThinColonSpace
1293     \ifFB@luatex@punct
1294         \FBcolonskip=\FBthinspace\relax
1295     \else
1296         \renewcommand*{\FBcolonspace}{\FBthinspace}%
1297     \fi
1298 \fi
```

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

```
1299 \@ifpackageloaded{numprint}%
1300 {\ifnprt@autolanguage
1301     \providecommand*{\npstylefrench}{}%
1302     \ifFBThinSpaceInFrenchNumbers
1303         \renewcommand*{\npstylefrench}%
1304             \npthousandsep{\,%
1305             \npdecimalsign{,%
1306             \npproductsign{\cdot}%
1307             \npunitseparator{\,%
1308             \npdegreeseperator{}%
1309             \nppercentseparator{\nprt@unitsep}%
```

```

1310     }%
1311   \else
1312     \renewcommand*\npstylefrench{%
1313       \npthousandsep{~}%
1314       \npdecimalsign{,}%
1315       \npproductsign{\cdot}%
1316       \npunitseparator{\,}%
1317       \npdegreeseparator{}%
1318       \nppercentseparator{\np@unitsep}%
1319     }%
1320   \fi
1321   \npaddtolanguage{french}{french}%
1322 \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.

```

1323 \ifFBFrenchSuperscripts
1324   \DeclareRobustCommand*\up*{\@ifstar{\FB@up@fake}{\fup}}%
1325 \else
1326   \DeclareRobustCommand*\up*{\@ifstar{\FB@up@fake}%
1327                                     {\textsuperscript}}%
1328 \fi

```

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

```

1329 \ifFBLowercaseSuperscripts
1330 \else
1331   \renewcommand*\FB@lc[1]{##1}%
1332 \fi

```

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

```

1333 \ifBFCustomiseFigTabCaptions
1334   \ifFB@koma
1335     \let\captionformat\CaptionSeparator
1336   \fi
1337   \@ifclassloaded{memoir}%
1338     {\captiondelim{\CaptionSeparator}}{}%
1339 \else

```

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

```

1340   \ifFB@koma
1341     \def\captionformat{\iflanguage{french}{ : }{: }}%
1342   \fi
1343   \@ifclassloaded{memoir}%
1344     {\captiondelim{\iflanguage{french}{ : }{: }}}{}%
1345 \fi

```

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

```

1346 \ifFBShowOptions

```

```

1347 \GenericWarning{* }{%
1348 * **** List of possible options for frenchb ****\MessageBreak
1349 [Default values between brackets when frenchb is loaded *LAST*]%
1350 \MessageBreak
1351 ShowOptions=true [false]\MessageBreak
1352 StandardLayout=true [false]\MessageBreak
1353 GlobalLayoutFrench=false [true]\MessageBreak
1354 StandardLists=true [false]\MessageBreak
1355 IndentFirst=false [true]\MessageBreak
1356 ReduceListSpacing=false [true]\MessageBreak
1357 ListOldLayout=true [false]\MessageBreak
1358 StandardItemizeEnv=true [false]\MessageBreak
1359 StandardEnumerateEnv=true [false]\MessageBreak
1360 StandardItemLabels=true [false]\MessageBreak
1361 ItemLabels=\textemdash, \textbullet,
1362 \protect\ding{43},... [\textendash]\MessageBreak
1363 ItemLabeli=\textemdash, \textbullet,
1364 \protect\ding{43},... [\textendash]\MessageBreak
1365 ItemLabelii=\textemdash, \textbullet,
1366 \protect\ding{43},... [\textendash]\MessageBreak
1367 ItemLabeliii=\textemdash, \textbullet,
1368 \protect\ding{43},... [\textendash]\MessageBreak
1369 ItemLabeliv=\textemdash, \textbullet,
1370 \protect\ding{43},... [\textendash]\MessageBreak
1371 FrenchFootnotes=false [true]\MessageBreak
1372 AutoSpaceFootnotes=false [true]\MessageBreak
1373 AutoSpacePunctuation=false [true]\MessageBreak
1374 OriginalTypewriter=true [false]\MessageBreak
1375 ThinColonSpace=true [false]\MessageBreak
1376 ThinSpaceInFrenchNumbers=true [false]\MessageBreak
1377 FrenchSuperscripts=false [true]\MessageBreak
1378 LowercaseSuperscripts=false [true]\MessageBreak
1379 PartNameFull=false [true]\MessageBreak
1380 SuppressWarning=true [false]\MessageBreak
1381 CustomiseFigTabCaptions=false [true]\MessageBreak
1382 og= <left quote character>, fg= <right quote character>%
1383 \MessageBreak
1384 *****%
1385 \MessageBreak\protect\frenchbsetup{ShowOptions}}
1386 \fi
1387 }

```

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.

```

1388 \AtBeginDocument{%
1389 \providecommand*\xspace{\relax}%

```

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

```

1390 \ifdefined\pdfstringdefDisableCommands

```

```

1391 \pdfstringdefDisableCommands{%
1392   \let\up\relax
1393   \let\fup\relax
1394   \let\degre\textdegree
1395   \let\degres\textdegree
1396   \def\ieme{e\xspace}%
1397   \def\iemes{es\xspace}%
1398   \def\ier{er\xspace}%
1399   \def\iers{ers\xspace}%
1400   \def\iere{re\xspace}%
1401   \def\ieres{res\xspace}%
1402   \def\FrenchEnumerate#1{#1\degre\space}%
1403   \def\FrenchPopularEnumerate#1{#1\degre)\space}%
1404   \def\No{N\degre\space}%
1405   \def\no{n\degre\space}%
1406   \def\Nos{N\degre\space}%
1407   \def\nos{n\degre\space}%
1408   \def\FB@og{\guillemotleft\space}%
1409   \def\FB@fg{\space\guillemotright}%
1410   \def\at{@}%
1411   \def\circonflexe{\string^}%
1412   \def\tild{\string~}%
1413   \let\bsc\textsc
1414 }%
1415 \fi

```

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

```

1416 \FBprocess@options

```

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

```

1417 \ifFB@luatex@punct
1418 \activate@luatexpunct
1419 \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; 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!

```

1420 \ifFBunicode
1421   \ifdefined\DeclareUTFcharacter
1422   \else
1423     \PackageWarning{frenchb.ldb}%
1424       {Add \protect\usepackage{fontspec} to the\MessageBreak
1425        preamble of your document,}%
1426   \fi
1427 \else
1428   \begingroup \newcommand{\FBOTone}{OT1}%
1429   \ifx\encodingdefault\FBOTone
1430     \PackageWarning{frenchb.ldb}%
1431       {OT1 encoding should not be used for French.%

```

```

1432      \MessageBreak
1433      Add \protect\usepackage[T1]{fontenc} to the
1434      preamble\MessageBreak of your document,}%
1435      \fi
1436      \endgroup
1437      \fi
1438 }

```

2.10 French lists

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

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

```

1439 \let\listORI\list
1440 \let\endlistORI\endlist
1441 \def\FB@listVsettings{%
1442     \setlength{\itemsep}{0.4ex plus 0.2ex minus 0.2ex}%
1443     \setlength{\parsep}{0.4ex plus 0.2ex minus 0.2ex}%
1444     \setlength{\topsep}{0.8ex plus 0.4ex minus 0.4ex}%
1445     \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`.

```

1446     \@tempdima=\parskip
1447     \addtolength{\topsep}{-\@tempdima}%
1448     \addtolength{\partopsep}{\@tempdima}%
1449 }
1450 \def\listFB#1#2{\listORI{#1}\FB@listVsettings #2}}
1451 \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 2.9).
- Vertical spacing between items, before and after the list, should be *null* with *no glue* added;

- In French the labels of itemize-lists are vertically aligned as follows:

Text starting at ‘parindent’
 ⇐ Leftmargin
 – first item. . .
 – first second level item
 – next one. . .
 – second item. . .

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

```

\FrenchLabelItem \newcommand*{\FrenchLabelItem}{\textendash}
\FrenchLabelItemii \newcommand*{\FrenchLabelItemii}{\FrenchLabelItem}
\FrenchLabelItemiii \newcommand*{\FrenchLabelItemiii}{\FrenchLabelItem}
\FrenchLabelItemiv \newcommand*{\FrenchLabelItemiv}{\FrenchLabelItem}
1455 \newcommand*{\FrenchLabelItemv}{\FrenchLabelItem}

```

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

```

1457 \newlength\listindentFB
1458 \setlength{\listindentFB}{-1pt}
1459 \newlength\labelwidthFB
1460 \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.

```

1461 \newlength\leftmarginFB
1462 \def\FB@listHsettings{%
1463   \leftmarginFB\labelwidthFB
1464   \advance\leftmarginFB \labelsep
1465   \leftmarginii\leftmarginFB
1466   \advance\leftmarginii \listindentFB
1467   \leftmarginiii\leftmarginFB
1468   \leftmarginiiii\leftmarginFB
1469   \leftmarginiv\leftmarginFB
1470   \leftmargin\cename leftmargin\romannumeral\the\@listdepth\endcename
1471 }

```

\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).

```

1472 \def\FB@itemizesettings{%
1473   \ifFBReduceListSpacing
1474     \setlength{\itemsep}{\z@}%
1475     \setlength{\parsep}{\z@}%

```

```

1476     \setlength{\topsep}{\z@}%
1477     \setlength{\partopsep}{\z@}%
1478     \@tempdima=\parskip
1479     \addtolength{\topsep}{-\@tempdima}%
1480     \addtolength{\partopsep}{\@tempdima}%
1481 \fi
1482 \settowidth{\labelwidth}{\csname\@itemitem\endcsname}%
1483 \ifFBListOldLayout
1484     \setlength{\leftmargin}{\labelwidth}%
1485     \addtolength{\leftmargin}{\labelsep}%
1486     \addtolength{\leftmargin}{\parindent}%
1487 \else
1488     \FB@listHsettings
1489 \fi
1490 }

```

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

```

1491 \def\itemizeFB{%
1492     \ifnum \@itemdepth >\thr@@\@toodeep\else
1493         \advance\@itemdepth\@ne
1494         \edef\@itemitem{\labelitem\romannumeral\the\@itemdepth}%
1495         \expandafter
1496             \listORI
1497             \csname\@itemitem\endcsname
1498             \FB@itemizesettings
1499     \fi
1500 }
1501 \let\enditemizeFB\endlistORI

1502 \def\labelitemsFB{%
1503     \let\labelitemi\Frlabelitemi
1504     \let\labelitemii\Frlabelitemii
1505     \let\labelitemiii\Frlabelitemiii
1506     \let\labelitemiv\Frlabelitemiv
1507     \ifdim\labelwidthFB<\z@
1508         \settowidth{\labelwidthFB}{\FrenchLabelItem}%
1509     \fi
1510     \ifdim\listindentFB<\z@
1511         \ifdim\parindent=\z@
1512             \setlength{\listindentFB}{1.5em}%
1513         \else
1514             \setlength{\listindentFB}{\parindent}%
1515         \fi
1516     \fi
1517 }

```

`\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`.

```

1518 \def\enumerateFB{%
1519   \ifnum \@enumdepth >\thr@@\toodeep\else
1520     \advance\@enumdepth\@ne
1521     \edef\@enumctr{enum\romannumeral\the\@enumdepth}%
1522     \expandafter
1523     \list
1524       \csname label\@enumctr\endcsname
1525       {\FB@listHsettings
1526         \usecounter\@enumctr\def\makelabel##1{\hss\llap{##1}}}%
1527   \fi
1528 }
1529 \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).

```

1530 \def\descriptionFB{%
1531   \list{}\FB@listHsettings
1532     \labelwidth\z@
1533     \itemindent-\leftmargin
1534     \ifnum\@listdepth=1
1535       \advance\itemindent by \listindentFB
1536     \fi
1537     \let\makelabel\descriptionlabel}%
1538 }
1539 \let\enddescriptionFB\endlistORI

```

\update@frenchlists \update@frenchlists will setup lists according to the options of \frenchbsetup{
\bbl@frenchlistlayout \bbl@frenchlistlayout
\bbl@nonfrenchlistlayout \bbl@nonfrenchlistlayout

```

1540 \def\update@frenchlists{%
1541   \ifFBReduceListSpacing \let\list\listFB \fi
1542   \ifFBStandardItemizeEnv
1543   \else \let\itemize\itemizeFB \fi
1544   \ifFBStandardItemLabels
1545   \else \labelitemsFB \fi
1546   \ifFBStandardEnumerateEnv
1547   \else \let\enumerate\enumerateFB \let\description\descriptionFB \fi
1548 }

```

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. 48.

```

1549 \def\FB@ufl{\relax}
1550 \def\bbl@frenchlistlayout{%
1551   \ifFBGlobalLayoutFrench
1552   \else
1553     \babel@save\list          \babel@save\itemize
1554     \babel@save\enumerate    \babel@save\description
1555     \babel@save\labelitemi   \babel@save\labelitemii
1556     \babel@save\labelitemiii \babel@save\labelitemiv
1557   \fi

```



```

1558 \FB@ufl
1559 }
1560 \def\bbl@nonfrenchlistlayout{%
1561 \ifFBGlobalLayoutFrench
1562 \update@frenchlists
1563 \fi
1564 }
1565 \FB@addto{extras}{\bbl@frenchlistlayout}
1566 \FB@addto{noextras}{\bbl@nonfrenchlistlayout}

```

2.11 French indentation of sections

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

```

1567 \def\bbl@frenchindent{%
1568 \ifFBGlobalLayoutFrench\else\babel@save\@afterindentfalse\fi
1569 \ifFBIndentFirst
1570 \let\@afterindentfalse\@afterindenttrue
1571 \@afterindenttrue
1572 \fi}
1573 \def\bbl@nonfrenchindent{%
1574 \ifFBGlobalLayoutFrench
1575 \ifFBIndentFirst
1576 \@afterindenttrue
1577 \fi
1578 \fi}
1579 \FB@addto{extras}{\bbl@frenchindent}
1580 \FB@addto{noextras}{\bbl@nonfrenchindent}

```

2.12 Formatting footnotes

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

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

```

1581 \AtBeginDocument{\@ifpackageloaded{bigfoot}%
1582 \PackageInfo{frenchb.lds}%
1583 {bigfoot package in use.\MessageBreak

```

```

1584         frenchb will NOT customise footnotes;\MessageBreak
1585         reported}}}%
1586     {\let\@footnotemarkORI\@footnotemark
1587      \def\@footnotemarkFB{\leavevmode\unskip\unkern
1588                          \,\@footnotemarkORI}%
1589      \ifFBAutoSpaceFootnotes
1590        \let\@footnotemark\@footnotemarkFB
1591      \fi}%
1592   }

```

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).

```

1593 \newcommand*{\dotFFN}{.}
1594 \newcommand*{\kernFFN}{\kern .5em}
1595 \newdimen\parindentFFN
1596 \parindentFFN=10in
1597 \def\ftnISSymbol{\@fnsymbol\c@footnote}
1598 \long\def\@makefntextFB#1{\ifx\thefootnote\ftnISSymbol
1599                             \@makefntextORI{#1}%
1600                             \else
1601                             \parindent=\parindentFFN
1602                             \rule\z@\footnotesep
1603                             \setbox\@tempboxa\hbox{\@thefnmark}%
1604                             \ifdim\wd\@tempboxa>\z@
1605                             \llap{\@thefnmark}\dotFFN\kernFFN
1606                             \fi #1
1607                             \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).

```

1608 \AtBeginDocument{\@ifpackageloaded{bigfoot}{}%
1609                  {\ifdim\parindentFFN<10in
1610                   \else
1611                     \parindentFFN=\parindent
1612                     \ifdim\parindentFFN<1.5em \parindentFFN=1.5em \fi
1613                   \fi
1614                   \let\@makefntextORI\@makefntext
1615                   \long\def\@makefntext#1{%
1616                     \ifFBFrenchFootnotes
1617                       \@makefntextFB{#1}%
1618                     \else

```

```

1619             \@makefntextORI{#1}%
1620             \fi}%
1621         }%
1622     }

```

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

```

1623 \newcommand*{\AddThinSpaceBeforeFootnotes}{\FBAutoSpaceFootnotesttrue}
1624 \newcommand*{\FrenchFootnotes}{\FBFrenchFootnotesttrue}
1625 \newcommand*{\StandardFootnotes}{\FBFrenchFootnotesfalse}

```

2.13 Clean up and exit

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

```

1626 \FBclean@on@exit
1627 \ldf@finish\CurrentOption

```

3 Change History

v2.0	General: <code>\parindentFFN</code> not changed if already defined (required by JA for <code>cah-gut.cls</code>).	58	and <code>StandardLayout</code> → <code>StandardLists</code>	41
	Added warning for OT1 encoding.	51		
	Footnotes are now printed by default ‘à la française’ for the whole document.	57		
	New command <code>\frenchbsetup</code> added for global customisation.	41		
	<code>\bsc</code> : <code>\hbox</code> dropped, replaced by <code>\kern0pt</code>	33		
	<code>\captionsfrench</code> : ‘Fig.’ changed to ‘Figure’ and ‘Tab.’ to ‘Table’.	36		
	<code>\datefrench</code> : 2 ‘relax’ added in <code>\today</code> ’s definition.	30		
	<code>\FBtextellipsis</code> : Added special case for LY1 encoding, see bug report from Bruno Voisin (2004/05/18).	40		
	<code>\nombre</code> : <code>\nombre</code> now requires <code>numprint.sty</code>	35		
v2.0b	General: Footnotes: Just do nothing (except warning) when the <code>bigfoot</code> package is loaded.	57		
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>	36		
	<code>\frenchbsetup</code> : Option <code>ThinSpaceInFrenchNumbers</code> added.	41		
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.	41		
v2.0e	<code>\frenchbsetup</code> : New option: <code>StandardLists</code>	41		
v2.0f	<code>\frenchbsetup</code> : <code>StandardLayout</code> option had no effect on lists. Test moved to <code>\FBprocess@options</code>	41		
	Two typos corrected in option <code>StandardLists</code> : <code>[false]</code> → <code>[true]</code>			
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>	41		
v2.1a	General: Command <code>\fup</code> added to produce better superscripts than <code>\textsuperscript</code>	31		
	<code>\datefrench</code> : <code>\today</code> changed (correction in 2.0 was wrong: <code>\today</code> was printed without spaces in toc).	30		
	<code>\frenchbsetup</code> : New option: <code>FrenchSuperscripts</code> to define <code>\up</code> as <code>\fup</code> or as <code>\textsuperscript</code>	41		
	New option: <code>LowercaseSuperscripts</code>	41		
v2.1b	General: Disable some commands in bookmarks.	51		
	<code>\fup</code> : Command <code>\fup</code> changed to use real superscripts from <code>fourier v. 1.6</code>	31		
v2.1c	General: Added commands <code>\Nos</code> and <code>\nos</code>	33		
	<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).	34		
	<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).	31		
v2.1d	General: Argument of <code>\ProvidesLanguage</code> changed above from ‘french’ to			

‘frenchb’ (otherwise <code>\listfiles</code> prints no date/version information). The real name of current language (french) as to be corrected before calling <code>\LdfInit</code> . 12	(suggested by JA). 58
Avoid warning “\end occurred when <code>\ifx ... incomplete</code> ” with LaTeX-2.09. 13	
v2.2a	v2.3c
<code>\frenchbsetup</code> : Default values of flags changed: default now means ‘StandardLayout’, they will be changed to ‘FrenchLayout’ AtEndOfPackage only if french is <code>\bbl@main@language</code> 41	General: Commands <code>\ttfamily</code> , <code>\rmfamily</code> and <code>\sffamily</code> have to be robust. Bug introduced in 2.3a, pointed out by Manuel Pégourié-Gonnard. 27
The global layout of the document is no longer changed when frenchb is not the last option of babel (<code>\bbl@main@language</code>). Suggested by Ulrike Fischer. 41	v2.3d
When frenchb is babel’s last option, French becomes the document’s main language, so GlobalLayout-French applies. 41	<code>\bbl@nonfrenchindent</code> : Bug correction: previous versions of frenchb set the flag <code>\if@afterindent</code> to false outside French which is correct for English but wrong for some languages like Spanish. Pointed out by Juan José Torrens. 57
<code>\fup</code> : <code>\newif</code> and <code>\newdimen</code> moved before <code>\ifLaTeXe</code> to avoid an error with plainTeX. 31	v2.3e
v2.3a	General: Execute <code>\AutoSpaceBeforeFDP</code> also in LaTeX to define <code>\FDP@colonspace</code> : needed for tex4ht, pointed out by MPG. . . . 26
General: <code>\NoAutoSpaceBeforeFDP</code> and <code>\AutoSpaceBeforeFDP</code> now set the flag <code>\ifFBAutoSpacePunctuation</code> accordingly (LaTeX only). 26	v2.4a
In LaTeX, frenchb no longer adds spaces before ‘high punctuation’ characters in computer code. Suggested by Yannis Haralambous. 27	General: <code>\PackageWarning</code> changed to <code>\FBWarning</code> (when bigfoot package in use). 57
<code>\frenchbsetup</code> : New option: OriginalTypewriter. Now frenchb switches to <code>\noautospace@beforeFDP</code> when a tt-font is in use. When OriginalTypewriter is set to true, frenchb behaves as in pre-2.3 versions. 41	<code>\CaptionSeparator</code> : <code>\PackageWarning</code> changed to <code>\FBWarning</code> (in case <code>\@makecaption</code> has been customised). <code>\FBWarning</code> is defined as <code>\PackageWarning</code> by default but can be made silent using <code>\frenchbsetup</code> , (suggested by MPG). 38
<code>\fup</code> : <code>\lowercase</code> changed to <code>\MakeLowercase</code> as the former doesn’t work for non ASCII characters in encodings like applemac, utf-8,... 31	<code>\frenchbsetup</code> : New option SuppressWarning. 41
v2.3b	<code>\ifFBXeTeX</code> : Added a new ‘if’ <code>\FBunicode</code> and some <code>\lccode</code> definitions to <code>\extrasfrench</code> and <code>\noextrasfrench</code> 14
General: New commands <code>\dotFFN</code> and <code>\kernFFN</code> for more flexibility	v2.4c
	General: In <code>\ttfamilyFB</code> , also cancel automatic spaces inside French guillemets coded as characters (see <code>\frenchbsetup</code>). 45
	In <code>\ttfamilyFB</code> , also cancel automatic spaces inside French guillemets entered as characters (see <code>\frenchbsetup</code>). 27
	v2.4d
	<code>\up</code> : Command <code>\up</code> defined with <code>\providecommand</code> instead of

	\newcommand as \up may be defined elsewhere (catalan.ldf). Bug pointed out by Felip Manyé i Ballester.	31		\ifFBXeTeX: Added two new ‘if’ \FBXeTeX and \FBLuaTeX as XeTeX and behave differently regarding the status of the French “apostrophe”.	14
v2.5a	General: \og and \fg do not print correctly in English when using XeTeX or LuaTeX, fixed by using \textquotedblleft and \textquotedblright defined above.	30	v2.5e	General: \pdfstringdefDisableCommands should redefine \FB@og and \FB@fg instead of \og and \fg so that it works also when quotes are entered as characters. Reported by Sébastien Gouezel.	51
	New command \NoAutoSpacing, suggested by MPG.	28	v2.5f	General: Changed definitions of \at, \circonflexe, \tild, \boi and \degre for Unicode based engines.	33
	Punctuation is no longer made active with XeTeX-based engines.	15		\FBtextellipsis: Unicode fonts also provide a ready made character for \textellipsis, let’s just use it (reported by Maxime Chupin, 2011/06/04).	40
	\captionsfrench: \emph deleted in \seename and \alsoname to match what is done for the other languages. Suggested by Marc Baudoin.	36	v2.5g	General: Add four \newif to control spacing of quotes (characters and control sequences).	22
	\FBthinspace: Define \FBthinspace for those who want to customise the width of the space before ; and co.	16		Redefine \degre, \degres \at \circonflexe and \tild for bookmarks. Add \fup also.	51
	\textquoteddblright: Change \guillemotleft and \guillemotright definitions for Unicode and provide definitions for \textquotedblleft and \textquotedblright. Insures correct printing of quotes by \og and \fg in French and outside.	28		When \ifFB@xetex@punct is true, ‘og’ and ‘fg’ options now set XeTeXcharclasses of these characters to \FB@guilo and \FB@guilf. Otherwise French quotes behave as normal characters (their XeTeX-charclass is 0).	45
v2.5b	General: Do not use the test \iflanguage{french} to check whether French is the main language or not, as it might be erroneously positive when English is the main language and no hyphenation patterns are available for French. In this case \l@french and \l@english are 0. Pointed out by Günter Milde.	42		\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.	23
v2.5c	General: The code meant for XeTeX also works for LuaTeX, we just need to change the test.	45	v2.5h	\degres: textcomp.sty has changed. The test about \M@TS1 is no longer relevant, let’s change it.	34
v2.5d	General: Moved the \newcount command outside \ifFB@xetex@punct ... \fi (it broke Plain formats). .	22	v2.5i	General: Temporary fix: as long as xeCJK.sty will not use	

<p><code>\newXeTeXintercharclass</code> to allocate its classes, we will have to define 3 fake classes. 22</p> <p><code>\FB@xetex@punct@french: xeCJK.sty</code> changes the <code>\XeTeXcharclass</code> of ASCII chars ‘ ’ ’ ’ ’ ’ ’ ’ ’ ’ ‘ ’ ’ ’ opening and closing single and double quotes. We set their class to 0 in French and reset their class to their original value when leaving French. See <code>\FB@xetex@punct@nonfrench</code> below. 23</p> <p>v2.5j</p> <p>General: Previous fix removed: bug fixed in <code>xeCJK.sty</code> version 3.0.4 (06-May-2012). 22</p> <p>v2.6a</p> <p>General: Bug correction: changing <code>\leftmargin</code> cannot be done only for itemize-lists: it messes up embedded enumerate lists. Pointed out by Denis Bitouzé. Lists have been completely redesigned in <code>frenchb</code> v. 2.6a. An option for backward compatibility is provided. 53</p> <p><code>\frenchbsetup</code>: New options <code>ListOldLayout</code>, <code>StandardItemizeEnv</code> and <code>StandardEnumerateEnv</code> (<code>CompactItemize</code> is deprecated). 41</p> <p><code>\FrenchLabelItem</code>: defaults to <code>\textendash</code> instead of <code>\textendash</code> up to v. 2.5k. 54</p> <p>v2.6b</p> <p><code>\descriptionFB</code>: Settings of <code>\FB@listHsettings</code> should apply to description lists too. 56</p> <p>v2.6c</p> <p>General: Dummy file <code>frenchb.cfg</code> is no longer generated from <code>frenchb.dtx</code>. 11</p> <p>No warning about <code>\@makecaption</code> for AMS classes. 39</p> <p>No warning about <code>\@makecaption</code> for koma-script classes. <code>\captionformat</code> customised in French. 39</p> <p>Warning added when the caption or floatrow package is loaded before <code>babel/frenchb</code>. 39</p>	<p><code>\CaptionSeparator</code>: Former <code>\CaptionSeparator</code> has been renamed as <code>\FBCaption@Separator</code>; Newif <code>\ifFBwarning@capsep</code> added. 38</p> <p>v2.6d</p> <p><code>\FBthinspace</code>: Rename <code>\Fthinspace</code> to <code>\FBthinspace</code> and <code>\Fcolonspace</code> to <code>\FBcolonspace</code> to avoid a conflict with <code>fournier.sty</code>. 16</p> <p>v2.6e</p> <p><code>\degres</code>: Refrain from redefining <code>\textdegree</code> from <code>latin1.def</code>, <code>applemac.def</code>, etc. as <code>\degres</code> because it loops in <code>hyperref</code>’s bookmarks. Pointed out by Eddy Flas on fctt. 34</p> <p>v2.6f</p> <p><code>\FB@itemizesettings: \labelwidth</code> must be reset, f.i. when an itemize list occurs inside environments based on <code>trivlist</code> which set <code>\labelwidth</code> to 0 (see proof environment in <code>amsthm.sty</code>). Bug pointed out by Julien Hauseux. . 54</p> <p>v2.6g</p> <p>General: U+00A0 (Unicode nobreakspace) and U+202F (Unicode nobreakthinspace) added to class <code>\FB@punctnul</code> to prevent <code>frenchb</code> from adding it’s own space before ‘high punctuation’ characters. 22</p> <p><code>\FB@itemizesettings</code>: Suppress all vertical spaces only if <code>ReduceListSpacing</code> is true. Pointed out by Pierre Willaime. 54</p> <p><code>\ifFBXeTeX</code>: lccode values for the French “apostrophe” are now the same for XeTeX and LuaTeX. ... 14</p> <p>v2.6h</p> <p>General: <code>\FG@og</code> and <code>\FG@fg</code> changed: former clumsy code removed. .. 45</p> <p>If <code>\@makecaption</code> is undefined, no warning. 39</p> <p>New class <code>\FB@guilnul</code> for characters U+00A0 (Unicode nobreakspace) and U+202F (Unicode nobreakthinspace), to prevent <code>frenchb</code> from adding spurious spaces inside quotes. 22</p>
---	---

\CaptionSeparator: No active cat-codes in \STD@makecaption's definition.	38	Test \ifXeTeX changed to \ifFBunicode and 'xltextra' changed to 'fontspec'.	52
v3.0a		\CaptionSeparator: Remove \CaptionSeparatorORI, use \babel@save instead.	38
General: \LdfInit checks \datefrench instead of \captionsfrench to avoid a conflict with papertex.cls which loads datetime.sty.	13	\captionsfrench: Take advantage of babel's \SetString commands for captionnames.	36
\bbl@nonfrenchguillemets deleted, use \babel@save instead.	30	\datefrench: Take advantage of babel's \SetString commands for \datefrench. Doesn't work with Plain (yet?).	30
Added explicit \FBguillesskip for LuaTeX.	29	\descriptionFB: Add \listindentFB to \itemindent. Suggested by Denis Bitouzé.	56
Definitions of \FB@og and \FB@fg now depend on punctuation handling (LuaTeX / XeTeX / active).	29	\extrasfrench: Take advantage of babel's \babel@savevariable to handle apostrophe's \lccode. ..	14
french.cfg will be loaded (if found) instead of frenchb.cfg. NO NEED for .cfg files in French anyway. .	59	\FBprocess@options: Changed option ThinColonSpace to make it work also with LuaTeX.	49
frenchb requires babel-3.9.	13	With koma-script and memoir class, customise \captionformat and \captiondelim.	50
In Plain, provide a substitute for \PackageWarning and \PackageInfo.	14	\FBthinskip: LuaTeX requires dimensions: two new skips \FBcolonskip and \FBthinskip.	16
Merging of \captionsfrenchb, \captionsfrançais with \captionsfrench deleted in favor of new babel 3.9 syntax. ...	37	\FBthinspace: \FBthinspace is no longer a kern but a skip (frenchb adds a nobreak penalty before it).	16
More informative, less TeXnical warning about \@makecaption. .	39	\frenchbsetup: New options OldFigTabCaptions and CustomiseFigTabCaptions.	41
New flag \ifFB@luatex@punct for 'high punctuation' management with LuaTeX engines.	15	v3.0b	
New handling of 'high punctuation' through callbacks with LuaTeX engines.	16	General: frenchb.lua was not found by Lua function dofile (not kpathsea aware). Call function kpse.find_file first, as suggested by Paul Gaborit.	21
No warning about \@makecaption for SMF classes. No warning either with LuaTeX or XeTeX engines.	39	Require luatexbase with LaTeX with case fontspec has not been loaded before babel.	16
Options processing completely reorganised.	41		
Support for options frenchb, français, canadien, acadian changed.	12		