

A Babel language definition file for French

frenchb.dtx v3.0c, 2014/04/18

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	8
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	27
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	37
2.8 Dots...	40
2.9 Setup options: keyval stuff	41
2.10 French lists	53
2.11 French indentation of sections	58
2.12 Formatting footnotes	58
2.13 Clean up and exit	60
3 Change History	61

1 The French language

The file `frenchb.dtx`¹, defines all the language definition macros for the French language.

Customisation for the French language is achieved following the book “Lexique des règles typographiques en usage à l’Imprimerie nationale” troisième édition (1994), ISBN-2-11-081075-0.

First version released: 1.1 (May 1996) as part of babel-3.6beta. Version 2.0a was released in February 2007 and version 3.0a in February 2014.

`frenchb` has been improved using helpful suggestions from many people, mainly from Jacques André, Michel Bovani, Thierry Bouche, Vincent Jalby and Denis Bitouzé. Thanks to all of them!

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

Changes between version 2.0 and v3.0c 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 v3.0c and was last revised on 2014/04/18.

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

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

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

Some commands are provided by `frenchb` to make typesetting easier:

1. French quotation marks can be entered using the commands `\og` and `\fg` which work in L^AT_EX_{2 ϵ} and PlainT_EX, their appearance depending on what is available to draw them; even if you use L^AT_EX_{2 ϵ} and T₁-encoding, you should refrain from entering them as `<<~French quotation~>>`: `\og` and `\fg` provide better horizontal spacing (controlled by `\FBguillspace`). If French quote characters are available on your keyboard, you can use them, to get proper spacing in L^AT_EX_{2 ϵ} see option `og=«, fg=»` p. 7.
`\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).

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

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 `TEXbook` p. 134). The command `\DecimalMathComma` makes the comma be an ordinary character *in French only* (no space added); as a counterpart, if `\DecimalMathComma` is active, an explicit space has to be added in lists and intervals: $[\![0, \ 1]\!]$, $(x, \ y)$. `\StandardMathComma` switches back to the standard behaviour of the comma.
8. A command `\nombre` was provided in 1.x versions to easily format numbers in slices of three digits separated either by a comma in English or with a space in French; `\nombre` is now mapped to `\numprint` from `numprint.sty`, see `numprint.pdf` for more information.
9. `frenchb` has been designed to take advantage of the `xspace` package if present: adding `\usepackage{xspace}` in the preamble will force macros like `\fg`, `\ier`, `\ieme`, `\dots`, ... to respect the spaces you type after them, for instance typing `'1\ier juin'` will print `'1er juin'` (no need for a forced space after `1\ier`).

1.2 Customisation

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

1.2.1 `\frenchbsetup{options}`

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

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

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

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

`ReduceListSpacing=false (true*)` ; `frenchb` reduces the values of the vertical spaces used in the *all* list environments in French (this includes

itemize, enumerate, description, but also abstract, quote, quotation and verse and possibly others). Setting this option to `false` reverts to the standard settings of the list environment.

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

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

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

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

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

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

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

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

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

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

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

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

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

can help when some footnotes are numbered with letters (inside minipages for instance).

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

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

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

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

`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 \text{évènement} alg\text{èbre}}
\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.

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

What's new in version 2.6?

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

frenchb is now compatible with the paralist package.

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

What's new in version 2.5?

The main change is that active characters are no longer used in French with (recent) XeTeX-based engines (they still are with TeX-based engines). All the functionalities (automatic insertion of missing spaces before ; ! ? or bare replacement of typed spaces with suitable unbreachable ones, tuning of the spaces width) remain available and the user interface is unchanged. The use of active characters is replaced by the \XeTeXinterchartoks mechanism (adapted from the polyglossia package).

A new command `\NoAutoSpacing` has been added. It should be used *inside a group* instead of `\shorthandoff{;:!?}` whenever active characters or automatic spacing of French punctuation or quote characters conflict with other packages; it is designed to work with TeX-, LuaTeX- and XeTeX-based engines. Bug corrections: `\frenchspacing` and `\nonfrenchspacing` are no longer messed up by `frenchb.ldf`.

What's new in version 2.4?

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

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

Better compatibility with the `enumitem` package.

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

What's new in version 2.3?

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

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

What's new in version 2.2?

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

What's new in version 2.1?

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

What's new in version 2.0?

Here is the list of all changes:

- Support for L^AT_EX-2.09 and for L^AT_EX 2_ε in compatibility mode has been dropped. This version is meant for L^AT_EX 2_ε and Plain based formats (like bplain). L^AT_EX 2_ε formats based on m^LT_EX are no longer supported either (plenty of good 8-bits fonts are available now, so T1 encoding should be preferred for typesetting in French). A warning is issued when OT1 encoding is in use at the `\begin{document}`.
- Customisation should now be handled only by command `\frenchbsetup{}`, `frenchb.cfg` (kept for compatibility) should no longer be used. See section 1.2 for the list of available options.
- Captions in figures and tables have changed in French: former abbreviations “Fig.” and “Tab.” have been replaced by full names “Figure” and “Table”. If this leads to formatting problems in captions, you can add the following two commands to your preamble (after loading babel) to get the former captions

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

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

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

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

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

2 The code

2.1 Initial setup

If `frenchb.ldf` was loaded with `babel's` options `francais` or `frenchb`, we make it behave as if `french` was specified. In Plain formats, `@` catcode is not 'letter'.

```
1 \chardef\atcatcode=\catcode'\@
2 \catcode'\@=11\relax
3 \def\bbl@tempa{francais}
4 \ifx\CurrentOption\bbl@tempa
5   \let\l@francais\l@french
6   \def\captionsfrancais{\captionsfrench}
7   \def\datefrancais{\datefrench}
8   \def\extrasfrancais{\extrasfrench}
9   \def\noextrasfrancais{\extrasfrench}
10  \def\CurrentOption{french}
11 \fi
12 \def\bbl@tempa{frenchb}
13 \ifx\CurrentOption\bbl@tempa
14   \let\l@frenchb\l@french
15   \def\captionsfrenchb{\captionsfrench}
16   \def\datefrenchb{\datefrench}
17   \def\extrasfrenchb{\extrasfrench}
18   \def\noextrasfrenchb{\extrasfrench}
19   \def\CurrentOption{french}
20 \fi
21 \catcode'\@=\atcatcode \let\atcatcode\relax
```

The macro `\LdfInit` takes care of preventing that this file is loaded more than once, checking the category code of the `@` sign, etc.

```
22 \LdfInit\CurrentOption\captionsfrench
```

Make sure that `\l@french` is defined (possibly as 0). `babel.def` now (3.9i) defines `\l@<language>` also for eTeX, LuaTeX and XeTeX formats which set `\lang@<language>`.

```
23 \def\FB@nopatterns{%
24   \ifx\l@nohyphenation\@undefined
25     \edef\bbl@nulllanguage{\string\language=0}%
26     \adddialect\l@french0
27   \else
28     \adddialect\l@french\l@nohyphenation
29     \edef\bbl@nulllanguage{\string\language=nohyphenation}%
30   \fi
31   \@nopatterns{French}}
32 \ifx\l@french\@undefined
33   \FB@nopatterns
34 \fi
```

`\ifLaTeXe` No support is provided for late L^AT_EX-2.09: issue a warning and exit if L^AT_EX-2.09 is in use. Plain is still supported.

```
35 \newif\ifLaTeXe
```

```

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

```

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

```

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

```

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

```

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

```

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

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

```

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

```

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

```

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

```

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

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

```

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

```

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

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

```

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

```

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

```

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

```

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

```

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

```

2.2 Punctuation

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

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

```

129 \newif\ifFBAutoSpaceGuilll \FBAutoSpaceGuillltrue

```

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

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

```

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

```

```

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

```

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

```

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

```

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

```

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

```

2.2.1 Punctuation with LuaTeX

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

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

```

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

```



```

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

```

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

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

```

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

```

Managing spacing after ‘«’ (U+00AB) and before ‘»’ (U+00BB) can be done by the way; we define two flags, FB_punct_left for characters requiring some space before them and FB_punct_right for ‘«’ which must be followed by some space.

```

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

```

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

```

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

```

189   [0xA0] = true,
190   [0x202F] = true}
191 local FB_guil_null =
192   {[0xA0] = true,
193    [0x202F] = true}

```

Local definitions for nodes:

```

194 local new_node = node.new
195 local copy_node = node.copy
196 local node_id = node.id
197 local GLUE = node_id("glue")
198 local GSPEC = node_id("glue_spec")

```

```

199 local GLYPH      = node_id("glyph")
200 local PENALTY     = node_id("penalty")
201 local nobreak     = new_node(PENALTY)
202 nobreak.penalty   = 10000
203 local insert_node_before = node.insert_before
204 local insert_node_after  = node.insert_after
205 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...

```

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

```

and a function to scale them for the current font:

```

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

```

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

```

233 local addDPspace   = luatexbase.attributes['FB@addDPspace']
234 local addGUILspace = luatexbase.attributes['FB@addGUILspace']
235 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*).

```

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

```

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

```

252     local glue = prev_id == GLUE and prev_subtype == 0
253     local glue_wd
254     if glue then
255       glue_spec = prev.spec
256       glue_wd = glue_spec.width
257     end
258     glue = glue and glue_wd > 0

```

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

```

259     if FB_punct_thin[char] or FB_punct_thick[char] then
260       local SBDP = has_attribute(item, addDPspace)
261       local fbglue
262       if FB_punct_thick[char] then
263         fbglue = new_glue_scaled(fid,colnwd,colnst,colnsh)
264       else
265         fbglue = new_glue_scaled(fid,thinwd,thinst,thinsh)
266       end
267       local auto =
268         SBDP and SBDP > 0 and
269         ((prev_char and not FB_punct_null[prev_char]) or
270         (not prev_char and (prev_id ~= 0 or prev_subtype ~= 3))
271       )

```

```

272         if glue or auto then
273             if glue then
274                 head = remove_node(head,prev,true)
275             end
276             insert_node_before(head, item, copy_node(nobreak))
277             insert_node_before(head, item, copy_node(fbglue))
278         end

```

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

```

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

```

290     if lang == FR and FB_punct_right[char] and SIG and SIG > 0 then
291         local next = item.next
292         local next_id, next_subtype, next_char
293         if next then
294             next_id = next.id
295             next_subtype = next.subtype
296             if next_id == GLYPH then
297                 next_char = next.char
298             end
299         end
300         local glue = next_id == GLUE and next_subtype == 0
301         if glue then
302             glue_spec = next.spec
303             glue_wd = glue_spec.width
304         end
305         glue = glue and glue_wd > 0
306         if glue or (next_char and not FB_guil_null[next_char]) then
307             if glue then
308                 head = remove_node(head,next,true)
309             end
310             local fid = item.font
311             local fbglue = new_glue_scaled(fid,guilwd,guilst,guilsh)

```

```

312         insert_node_after(head, item, copy_node(fbglue))
313         insert_node_after(head, item, copy_node(nobreak))
314     end
315 end
316 end
317 return head
318 end
319 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.

```

320 \ifFB@luatexpunct
321   \newcommand*{\FB@luatexpunct@french}{%
322     \ifx\shorthandoffORI\undefined
323       \let\shorthandonORI\shorthandon
324       \let\shorthandoffORI\shorthandoff
325     \fi
326     \def\shorthandoff##1{%
327       \ifx\PackageWarning\undefined
328         \fb@warning{\noexpand\shorthandoff{;:!?} is helpless with
329           LuaTeX,\, use \noexpand\NoAutoSpacing
330           *inside a group* instead.}%
331       \else
332         \PackageWarning{frenchb.ldf}{\protect\shorthandoff{;:!?} is
333           helpless with LuaTeX,\MessageBreak use \protect\NoAutoSpacing
334           \space *inside a group* instead;\MessageBreak reported}%
335       \fi}%
336     \def\shorthandon##1{%
337   }
338   \newcommand*{\FB@luatexpunct@nonfrench}{%
339     \ifx\shorthandoffORI\undefined
340     \else
341       \let\shorthandon\shorthandonORI
342       \let\shorthandoff\shorthandoffORI
343     \fi
344   }
345   \FB@addto{extras}{\FB@luatexpunct@french}
346   \FB@addto{noextras}{\FB@luatexpunct@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).

```

347 \def\activate@luatexpunct{%
348   \directlua{%
349     FR = \the\l@french
350     local path = kpse.find_file("frenchb.lua", "lua")

```

```

351     if path then
352       local f = dofile(path)
353       luatexbase.add_to_callback("pre_linebreak_filter",
354         f, "frenchb.french_punctuation",1)
355       luatexbase.add_to_callback("hpack_filter",
356         f, "frenchb.french_punctuation",1)
357     else
358       texio.write_nl('')
359       texio.write_nl('*****')
360       texio.write_nl('Error: frenchb.lua not found.')
361       texio.write_nl('*****')
362       texio.write_nl('')
363     end
364   }%
365 }
366 \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.

```

367 \newcount\FB@interchartokenstateORI
368 \ifFB@xetex@punct
369   \ifLaTeXe
370     \PackageInfo{frenchb.ldf}{No need for active punctuation characters%
371       \MessageBreak with this version of XeTeX!%
372       \MessageBreak reported}
373   \else
374     \fb@info{No need for active punctuation characters\\
375       with this version of XeTeX!}
376   \fi

```

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

```

377   \ifx\@for\@undefined
378     \def\@nnil{\@nil}%

```

```

379 \def\@empty{}%
380 \def\@fornoop#1\@#2#3{%
381 \long\def\@for#1:=#2\do#3{%
382 \expandafter\def\expandafter\@fortmp\expandafter{#2}%
383 \ifx\@fortmp\@empty \else
384 \expandafter\@forloop#2,\@nil,\@nil\@#1{#3}\fi}%
385 \long\def\@forloop#1,#2,#3\@#4#5{\def#4{#1}\ifx #4\@nnil \else
386 #5\def#4{#2}\ifx #4\@nnil \else#5\@iforloop #3\@#4{#5}\fi\fi}%
387 \long\def\@iforloop#1,#2\@#3#4{\def#3{#1}\ifx #3\@nnil
388 \expandafter\@fornoop \else
389 #4\relax\expandafter\@iforloop\fi#2\@#3{#4}}%
390 \def\@tfor#1:={\@tfor#1 }%
391 \long\def\@tfor#1#2\do#3{\def\@fortmp{#2}\ifx\@fortmp\space\else
392 \@tforloop#2\@nil\@nil\@#1{#3}\fi}%
393 \long\def\@tforloop#1#2\@#3#4{\def#3{#1}\ifx #3\@nnil
394 \expandafter\@fornoop \else
395 #4\relax\expandafter\@tforloop\fi#2\@#3{#4}}%
396 \fi

```

Six new character classes are defined for frenchb.

```

397 \newXeTeXintercharclass\FB@punctthick
398 \newXeTeXintercharclass\FB@punctthin
399 \newXeTeXintercharclass\FB@punctnul
400 \newXeTeXintercharclass\FB@guilo
401 \newXeTeXintercharclass\FB@guilf
402 \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.

```

403 \def\FB@charclassesORI{}
404 \def\empty{}
405 \def\FB@parse#1,#2\endparse{\def\FB@class{#1}%
406 \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.

```

407 \newcommand*\FB@xetex@punct@french{%

```

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

```

408 \ifx\FB@charclassesORI\empty
409 \FB@interchartokenstateORI=\XeTeXinterchartokenstate
410 \@for\FB@char:={'\:,\';,\!\,'?,"AB,"BB,%
411 '\(,\[,\{,\,,\.,\-,\'\'',\],\'\'',%
412 '\%, "22, "27, "60, "2019, "A0, "202F}\do
413 {\edef\FB@charclassesORI{\FB@charclassesORI%
414 \theXeTeXcharclass\FB@char,}}%
415 \let\shorthandonORI\shorthandon
416 \let\shorthandoffORI\shorthandoff
417 \fi

```

Set the classes and interactions between classes.

```

418 \XeTeXinterchartokenstate=1
419 \XeTeXcharclass '\: = \FB@punctthick
420 \XeTeXinterchartoks \z@ \FB@punctthick = {%
421     \ifhmode\FDP@colonspace\fi}%
422 \XeTeXinterchartoks \FB@guilf \FB@punctthick = {%
423     \FDP@colonspace}%
424 \XeTeXinterchartoks 255 \FB@punctthick = {%
425     \ifhmode\unskip\penalty\@M\FBcolonspace\fi}%
426 \@for\FB@char:={'\;,'!\,'?}\do
427     {\XeTeXcharclass\FB@char=\FB@punctthin}%
428 \XeTeXinterchartoks \z@ \FB@punctthin = {%
429     \ifhmode\FDP@thinspace\fi}%
430 \XeTeXinterchartoks \FB@guilf \FB@punctthin = {%
431     \FDP@thinspace}%
432 \XeTeXinterchartoks 255 \FB@punctthin = {%
433     \ifhmode\unskip\penalty\@M\FBthinspace\fi}%
434 \XeTeXinterchartoks \FB@guilo \z@ = {%
435     \ifFBAutoSpaceGuill\FBguillspace\fi}%
436 \XeTeXinterchartoks \FB@guilo 255 = {%
437     \ifFBAutoSpaceGuill\FBguillspace\ignorespaces\fi}%
438 \XeTeXinterchartoks \z@ \FB@guilf = {%
439     \ifFBAutoSpaceGuill\FBguillspace\fi}%
440 \XeTeXinterchartoks \FB@punctthin \FB@guilf = {%
441     \ifFBAutoSpaceGuill\FBguillspace\fi}%
442 \XeTeXinterchartoks 255 \FB@guilf = {%
443     \ifFBAutoSpaceGuill\unskip\FBguillspace\fi}%

```

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

```

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

```

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

```

446 \@for\FB@char:={'\{, '\,, '\., '\- , '\), '\], '\}, '\%, %
447     "22, "27, "60, "2019}\do
448     {\XeTeXcharclass\FB@char=\z@}%

```

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

```

449 \def\shorthandoff##1{%
450     \ifx\PackageWarning\@undefined
451         \fb@warning{\noexpand\shorthandoff{;:!?} is helpless with
452             XeTeX,\, use \noexpand\NoAutoSpacing
453             *inside a group* instead.}%
454     \else
455         \PackageWarning{frenchb.lda}{\protect\shorthandoff{;:!?} is
456             helpless with XeTeX,\MessageBreak use \protect\NoAutoSpacing
457             \space *inside a group* instead;\MessageBreak reported}%
458     \fi}%
459 \def\shorthandon##1{%
460 }

```


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

```

461 \newcommand*\FB@xetex@punct@nonfrench{%
462   \ifx\FB@charclassesORI\empty
463   \else
464     \@for\FB@char:={'\:,\;\;,\!\;,\?,"AB,"BB,%
465                   '\(,\[,'\{,\,,'\.,'\-,\'),'\],'\},%
466                   '\%, "22,"27,"60,"2019,"A0,"202F}\do
467     {\expandafter\FB@parse\FB@charclassesORI\endparse
468      \XeTeXcharclass\FB@char=\FB@class}%
469     \def\FB@charclassesORI{%
470     \XeTeXinterchartokenstate=\FB@interchartokenstateORI
471     \let\shorthandon\shorthandonORI
472     \let\shorthandoff\shorthandoffORI
473   \fi
474 }
475 \FB@addto{extras}{\FB@xetex@punct@french}
476 \FB@addto{noextras}{\FB@xetex@punct@nonfrench}

```

End of specific code for punctuation with modern XeTeX engines.

477 \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.

```

478 \ifFB@active@punct
479   \initiate@active@char{:}%
480   \initiate@active@char{;}%
481   \initiate@active@char{!}%
482   \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`.

```

483 \declare@shorthand{french}{;}{;%
484   \ifhmode
485     \ifdim\lastskip>\z@
486       \unskip\penalty\M\FBthinspace
487     \else
488       \FDP@thinspace
489     \fi
490   \fi

```

Now we can insert a ; character.

```

491   \string;}

```

The next three definitions are very similar.

```

492 \declare@shorthand{french}{!}{%
493     \ifhmode
494         \ifdim\lastskip>\z@
495             \unskip\penalty\@M\FBthinspace
496         \else
497             \FDP@thinspace
498         \fi
499     \fi
500     \string!}
501 \declare@shorthand{french}{?}{%
502     \ifhmode
503         \ifdim\lastskip>\z@
504             \unskip\penalty\@M\FBthinspace
505         \else
506             \FDP@thinspace
507         \fi
508     \fi
509     \string?}
510 \declare@shorthand{french}{:}{%
511     \ifhmode
512         \ifdim\lastskip>\z@
513             \unskip\penalty\@M\FBcolonspace
514         \else
515             \FDP@colonspace
516         \fi
517     \fi
518     \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.

```

519 \declare@shorthand{system}{:}{\string:}
520 \declare@shorthand{system}{!}{\string!}
521 \declare@shorthand{system}{?}{\string?}
522 \declare@shorthand{system}{;}{\string;}
523 %}

```

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

```

524 \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.

```

525     \bbl@activate{:}\bbl@activate{;}%
526     \bbl@activate{!}\bbl@activate{?}%
527 }
528 \FB@addto{noextras}{%
529     \bbl@deactivate{:}\bbl@deactivate{;}%
530     \bbl@deactivate{!}\bbl@deactivate{?}%
531 }

```

532 \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.

533 \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).

```
534 \def\autospace@beforeFDP{%
535     \ifFB@luatex@punct\FB@addDPspace=1 \fi
536     \def\FDP@thinspace{\penalty\@M\FBthinspace}%
537     \def\FDP@colonspace{\penalty\@M\FBcolonspace}}
538 \def\noautospace@beforeFDP{%
539     \ifFB@luatex@punct\FB@addDPspace=0 \fi
540     \let\FDP@thinspace\@empty
541     \let\FDP@colonspace\@empty}
542 \ifLaTeXe
543     \def\AutoSpaceBeforeFDP{\autospace@beforeFDP
544                             \FBAutoSpacePunctuationtrue}
545     \def\NoAutoSpaceBeforeFDP{\noautospace@beforeFDP
546                               \FBAutoSpacePunctuationfalse}
547     \AtEndOfPackage{\AutoSpaceBeforeFDP}
548 \else
549     \let\AutoSpaceBeforeFDP\autospace@beforeFDP
550     \let\NoAutoSpaceBeforeFDP\noautospace@beforeFDP
551     \AutoSpaceBeforeFDP
552 \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.

553 \ifLaTeXe

```

554 \DeclareRobustCommand\ttfamilyFB{%
555     \FBAutoSpaceGuillfalse
556     \ifFB@luatex@punct\FB@addGUILspace=0 \fi
557     \noautospace@beforeFDP\ttfamilyORI}%
558 \DeclareRobustCommand\rmfamilyFB{%
559     \FBAutoSpaceGuilltrue
560     \ifFB@luatex@punct\FB@addGUILspace=1 \fi
561     \ifFBAutoSpacePunctuation
562         \autospace@beforeFDP
563     \else
564         \noautospace@beforeFDP
565     \fi
566     \rmfamilyORI}%
567 \DeclareRobustCommand\sffamilyFB{%
568     \FBAutoSpaceGuilltrue
569     \ifFB@luatex@punct\FB@addGUILspace=1 \fi
570     \ifFBAutoSpacePunctuation
571         \autospace@beforeFDP
572     \else
573         \noautospace@beforeFDP
574     \fi
575     \sffamilyORI}%
576 \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.

```

577 \newcommand*{\NoAutoSpacing}{\FBAutoSpaceGuillfalse
578     \ifFB@active@punct\shorthandoff{;:!?}\fi
579     \ifFB@xetex@punct\XeTeXinterchartokenstate=0 \fi
580     \ifFB@luatex@punct\FB@addDPspace=0 \FB@addGUILspace=0 \fi
581 }

```

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`.

```

582 \newcommand*{\og}{\@empty}
583 \newcommand*{\fg}{\@empty}

```

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

and LuaLaTeX, `\guillemotleft` and `\guillemotright` are defined by package `xunicode` loaded by `fontspec`.

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

```

584 \ifLaTeXe
585 \else
586   \ifFBunicode
587     \def\guillemotleft{\char"00AB}
588     \def\guillemotright{\char"00BB}
589     \def\textquotedblleft{\char"201C}
590     \def\textquotedblright{\char"201D}
591   \else
592     \def\guillemotleft{\leavevmode\raise0.25ex
593                       \hbox{$\scriptscriptstyle\ll$}}
594     \def\guillemotright{\raise0.25ex
595                        \hbox{$\scriptscriptstyle\gg$}}
596     \def\textquotedblleft{‘}
597     \def\textquotedblright{’}
598   \fi
599   \let\xspace\relax
600 \fi

```

The next step is to provide correct spacing after `\guillemotleft` and before `\guillemotright`: a space precedes and follows quotation marks but no line break is allowed neither *after* the opening one, nor *before* the closing one. `\FBguillspace` which does the spacing, has been fine tuned by Thierry Bouche to 80% of an inter-word space but with reduced stretchability. French quotes (including spacing) are printed by `\FB@og` and `\FB@fg`, the expansion of the top level commands `\og` and `\og` is different in and outside French. We'll try to be smart to users of David Carlisle's `xspace` package: if this package is loaded there will be no need for `{}` or `\` to get a space after `\fg`, otherwise `\xspace` will be defined as `\relax` (done at the end of this file).

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

```

601 \newskip\FBguillskip
602 \FBguillskip=2.664pt plus 0.500pt minus 0.888pt \relax
603 \newcommand*{\FBguillspace}{\penalty\@M\hskip.8\fontdimen2\font
604                               plus.3\fontdimen3\font
605                               minus.8\fontdimen4\font}

```

`\FBguillspace` is not used with LuaTeX.

```

606 \ifFB@luatex@punct
607   \DeclareRobustCommand*{\FB@og}{\leavevmode
608     \bgroup\FB@addGUllspace=1 \guillemotleft\egroup}
609   \DeclareRobustCommand*{\FB@fg}{\ifdim\lastskip>\z@\unskip\fi
610     \bgroup\FB@addGUllspace=1 \guillemotright\egroup\xspace}
611 \fi

```

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

```

612 \ifFB@xetex@punct
613   \DeclareRobustCommand*{\FB@og}{\leavevmode
614     \bgroup\FBAutoSpaceGuillfalse\guillemotleft\egroup
615     \FBguillspace}
616   \DeclareRobustCommand*{\FB@fg}{\ifdim\lastskip>z@\unskip\fi
617     \FBguillspace
618     \bgroup\FBAutoSpaceGuillfalse\guillemotright\egroup\xspace}
619 \fi
620 \ifFB@active@punct
621   \DeclareRobustCommand*{\FB@og}{\leavevmode
622     \guillemotleft
623     \FBguillspace}
624   \DeclareRobustCommand*{\FB@fg}{\ifdim\lastskip>z@\unskip\fi
625     \FBguillspace
626     \guillemotright\xspace}
627 \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.

```

628 \ifLaTeXe
629   \def\bbl@frenchguillemets{\renewcommand*{\og}{\FB@og}%
630     \renewcommand*{\fg}{\FB@fg}}
631   \renewcommand*{\og}{\textquotedblleft}
632   \renewcommand*{\fg}{\ifdim\lastskip>z@\unskip\fi \textquotedblright}
633 \else
634   \def\bbl@frenchguillemets{\let\og\FB@og
635     \let\fg\FB@fg}
636   \def\og{\textquotedblleft}
637   \def\fg{\ifdim\lastskip>z@\unskip\fi\textquotedblright}
638 \fi
639 \FB@addto{extras}{\babel@save\og \babel@save\fg \bbl@frenchguillemets}

```

2.4 Date in French

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

```

640 \ifLaTeXe
641   \def\BabelLanguages{french,acadian}
642   \StartBabelCommands*{\BabelLanguages}{date}
643     [unicode, fontenc=EU1 EU2, charset=utf8]
644     \SetString\monthiiname{février}
645     \SetString\monthviiiname{août}
646     \SetString\monthxiiname{décembre}
647   \StartBabelCommands*{\BabelLanguages}{date}
648     \SetStringLoop{month#1name}{%
649       janvier,f\'evrier,mars,avril,mai,juin,juillet,%
650       ao\^ut,septembre,octobre,novembre,d\'ecembre}

```

```

651 \SetString\today{{\number\day}\ifnum1=\day {\ier}\fi\space
652 \csname month\romannumeral\month name\endcsname \space
653 \number\year
654 }
655 \EndBabelCommands
656 \else
657 \ifFBunicode
658 \@namedef{date\CurrentOption}{%
659 \def\today{{\number\day}\ifnum1=\day {\ier}\fi \space
660 \ifcase\month
661 \or janvier\or février\or mars\or avril\or mai\or
662 juin\or juillet\or août\or septembre\or
663 octobre\or novembre\or décembre\fi
664 \space \number\year}}
665 \else
666 \@namedef{date\CurrentOption}{%
667 \def\today{{\number\day}\ifnum1=\day {\ier}\fi \space
668 \ifcase\month
669 \or janvier\or f'evrier\or mars\or avril\or mai\or
670 juin\or juillet\or ao^ut\or septembre\or
671 octobre\or novembre\or d'ecembre\fi
672 \space \number\year}}
673 \fi
674 \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).

```

675 \newif\ifFB@poorman
676 \newdimen\FB@Mht
677 \ifLaTeXe
678 \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{ }.

```

679 \newcommand*\FBsupR{-0.12}
680 \newcommand*\FBsupS{0.65}
681 \newcommand*\FB@lc[1]{\MakeLowercase{#1}}
682 \DeclareRobustCommand*\FB@up@fake[1]{%
683   \settoheight{\FB@Mht}{M}%
684   \addtolength{\FB@Mht}{\FBsupR \FB@Mht}%
685   \addtolength{\FB@Mht}{-\FBsupS ex}%
686   \raisebox{\FB@Mht}{\scalefont{\FBsupS}{\FB@lc{#1}}}%
687 }

```

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.

```

688 \def\FB@split#1#2#3#4\@nil{\def\FB@firstthree{#1#2#3}%
689   \def\FB@suffix{#4}}
690 \def\FB@x{x}
691 \def\FB@j{j}
692 \DeclareRobustCommand*\FB@up[1]{%
693   \bgroup \FB@poormantrue
694   \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.

```

695   \edef\reserved@a{\lowercase{%
696     \noexpand\IfFileExists{\f@encoding\FB@firstthree -sup.fd}}}%
697   \reserved@a
698   {\ifx\FB@suffix\FB@x \FB@poormanfalse\fi
699     \ifx\FB@suffix\FB@j \FB@poormanfalse\fi
700     \ifFB@poorman \FB@up@fake{#1}%
701     \else \FB@up@real{#1}%
702     \fi}%
703   {\FB@up@fake{#1}}%
704   \egroup}

```

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


```

705 \newcommand*{\FB@up@real}[1]{\bgroup
706     \fontfamily{\FB@firstthree -sup}\selectfont \FB@lc{#1}\egroup}
\fbup is defined as \FB@up unless \realsuperscript is defined by realscripts.sty.
707 \DeclareRobustCommand*{\fbup}[1]{%
708     \ifx\realsuperscript\undefined
709         \FB@up{#1}%
710     \else
711         \bgroup\let\fakesuperscript\FB@up@fake
712             \realsuperscript{\FB@lc{#1}}\egroup
713     \fi}

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

Poor man's definition of \up for Plain.
715 \else
716 \providecommand*{\up}[1]{\leavevmode\raiselex\hbox{\sevenrm #1}}
717 \fi

```

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

```

\ier 718 \def\ieme{\up{\lowercase{e}}\xspace}
\iere 719 \def\iemes{\up{\lowercase{es}}\xspace}
\iemes 720 \def\ier{\up{\lowercase{er}}\xspace}
\iers 721 \def\iers{\up{\lowercase{ers}}\xspace}
\ieres 722 \def\iere{\up{\lowercase{re}}\xspace}
723 \def\ieres{\up{\lowercase{res}}\xspace}

```

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

```

\no 724 \newcommand*{\FrenchEnumerate}[1]{%
\nos 725     #1\up{\lowercase{o}}\kern+.3em}
\nos 726 \newcommand*{\FrenchPopularEnumerate}[1]{%
\primo 727     #1\up{\lowercase{o}}\kern+.3em}

```

\fprimo) Typing \primo should result in '1°',

```

728 \def\primo{\FrenchEnumerate1}
729 \def\secundo{\FrenchEnumerate2}
730 \def\tertio{\FrenchEnumerate3}
731 \def\quarto{\FrenchEnumerate4}

while typing \fprimo) gives '1°) .
732 \def\fprimo){\FrenchPopularEnumerate1}
733 \def\fsecundo){\FrenchPopularEnumerate2}
734 \def\ftertio){\FrenchPopularEnumerate3}
735 \def\fquarto){\FrenchPopularEnumerate4}

```

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

```

736 \DeclareRobustCommand*{\No}{N\up{\lowercase{o}}\kern+.2em}
737 \DeclareRobustCommand*{\no}{n\up{\lowercase{o}}\kern+.2em}
738 \DeclareRobustCommand*{\Nos}{N\up{\lowercase{os}}\kern+.2em}
739 \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}`.

```
740 \DeclareRobustCommand*\bsc}[1]{\leavevmode\begingroup\kern0pt
741                                     \scshape #1\endgroup}
742 \ifLaTeXe\else\let\scshape\relax\fi
```

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

```
743 \ifBUnicode
744   \newcommand*\at{{\char"0040}}
745   \newcommand*\circonflexe{{\char"005E}}
746   \newcommand*\tild{{\char"007E}}
747   \newcommand*\boi{{\textbackslash}}
748   \newcommand*\degre{{\char"00B0}}
749 \else
750   \ifLaTeXe
751     \DeclareTextSymbol{\at}{T1}{64}
752     \DeclareTextSymbol{\circonflexe}{T1}{94}
753     \DeclareTextSymbol{\tild}{T1}{126}
754     \DeclareTextSymbolDefault{\at}{T1}
755     \DeclareTextSymbolDefault{\circonflexe}{T1}
756     \DeclareTextSymbolDefault{\tild}{T1}
757     \DeclareRobustCommand*\boi{{\textbackslash}}
758     \DeclareRobustCommand*\degre{{\r{}}}
759   \else
760     \def\T@one{T1}
761     \ifx\fontencoding\T@one
762       \newcommand*\degre{{\char6}}
763     \else
764       \newcommand*\degre{{\char23}}
765     \fi
766     \newcommand*\at{{\char64}}
767     \newcommand*\circonflexe{{\char94}}
768     \newcommand*\tild{{\char126}}
769     \newcommand*\boi{{\backslash}}
770   \fi
771 \fi
```

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

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

```

772 \ifLaTeXe
773   \newcommand*{\degrees}{\degree}
774   \ifFBunicode
775     \DeclareRobustCommand*{\degrees}{\degree}
776   \else
777     \def\Warning@degree@TSone{%
778       \PackageWarning{frenchb.lfd}{%
779         Degrees would look better in TS1-encoding:%
780         \MessageBreak add \protect
781         \usepackage{textcomp} to the preamble.%
782         \MessageBreak Degrees used}}
783     \AtBeginDocument{\ifx\DeclareEncodingSubset\@undefined
784       \DeclareRobustCommand*{\degrees}{%
785         \leavevmode\hbox to 0.3em{\hss\degree\hss}%
786         \Warning@degree@TSone
787         \global\let\Warning@degree@TSone\relax}%
788       \else
789         \DeclareRobustCommand*{\degrees}{%
790           \hbox{\UseTextSymbol{TS1}{\textdegree}}}%
791       \fi
792     }
793   \fi
794 \else
795   \newcommand*{\degrees}{%
796     \leavevmode\hbox to 0.3em{\hss\degree\hss}}
797 \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.

```

798 \newcount\std@mcc
799 \newcount\dec@mcc
800 \std@mcc=\mathcode'\,
801 \dec@mcc=\std@mcc
802 \@tempcnta=\std@mcc
803 \divide\@tempcnta by "1000
804 \multiply\@tempcnta by "1000
805 \advance\dec@mcc by -\@tempcnta
806 \newcommand*{\DecimalMathComma}{\iflanguage{french}%
807   {\mathcode'\,=\dec@mcc}}%
808   \FB@addto{extras}{\mathcode'\,=\dec@mcc}%

```

```

809 }
810 \newcommand*{\StandardMathComma}{\mathcode'\,,=\std@mcc
811   \FB@addto{extras}{\mathcode'\,,=\std@mcc}%
812 }
813 \FB@addto{noextras}{\mathcode'\,,=\std@mcc}

```

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

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

```

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

```

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

```

816 \let\FBstop@here\relax
817 \def\FBclean@on@exit{\let\ifLaTeXe\undefined
818                      \let\LaTeXettrue\undefined
819                      \let\LaTeXefalse\undefined}
820 \ifx\magnification\@undefined
821 \else
822   \def\FBstop@here{\ifFB@luatex@punct
823                     \activate@luatexpunct
824                     \fi
825                     \FBclean@on@exit
826                     \ldf@quit\CurrentOption\endinput}
827 \fi
828 \FBstop@here

```

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

```

829 \renewcommand*{\nombre}[1]{\Warning@nombre\numprint{#1}}
830 \newcommand*{\Warning@nombre}{%
831   \ifdefined\numprint
832   \else
833     \PackageWarning{frenchb.ldf}{%
834       \protect\nombre\space now relies on package numprint.sty,%
835       \MessageBreak add \protect
836       \usepackage[autolanguage]{numprint}\MessageBreak
837       to your preamble *after* loading babel,\MessageBreak
838       see file numprint.pdf for more options.\MessageBreak
839       \protect\nombre\space called}%
840   \global\let\Warning@nombre\relax
841   \fi

```

842 }

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).

```
843 \StartBabelCommands*{\BabelLanguages}{captions}
844     [unicode, fontenc=EU1 EU2, charset=utf8]
845     \SetString{\refname}{Références}
846     \SetString{\abstractname}{Résumé}
847     \SetString{\prefacename}{Préface}
848     \SetString{\contentsname}{Table des matières}
849     \SetString{\ccname}{Copie à }
850     \SetString{\proofname}{Démonstration}
851     \SetStringLoop{ordinal#1}{%
852         Première,Deuxième,Troisième,Quatrième,Cinquième,%
853         Sixième,Septième,Huitième,Neuvième,Dixième,Onzième,%
854         Douzième,Treizième,Quatorzième,Quinzième,Seizième,%
855         Dix-septième,Dix-huitième,Dix-neuvième,Vingtième}
856 \StartBabelCommands*{\BabelLanguages}{captions}
857     \SetString{\refname}{R\`ef\`erences}
858     \SetString{\abstractname}{R\`esum\`e}
859     \SetString{\bibname}{Bibliographie}
860     \SetString{\prefacename}{Pr\`eface}
861     \SetString{\chaptername}{Chapitre}
862     \SetString{\appendixname}{Annexe}
863     \SetString{\contentsname}{Table des mati\`eres}
864     \SetString{\listfigurename}{Table des figures}
865     \SetString{\listtablename}{Liste des tableaux}
866     \SetString{\indexname}{Index}
867     \SetString{\figurename}{{\scshape Figure}}
868     \SetString{\tablename}{{\scshape Table}}
869     \SetString{\pagename}{page}
870     \SetString{\seename}{voir}
871     \SetString{\alsoname}{voir aussi}
872     \SetString{\enclname}{P.~J. }
873     \SetString{\ccname}{Copie \`a }
874     \SetString{\headtoname}{}
875     \SetString{\proofname}{D\`emonstration}
876     \SetString{\glossaryname}{Glossaire}
877     \SetStringLoop{ordinal#1}{%
878         Premi\`ere,Deuxi\`eme,Troisi\`eme,Quatri\`eme,Cinqui\`eme,%
879         Sixi\`eme,Septi\`eme,Huiti\`eme,Neuvi\`eme,Dixi\`eme,Onzi\`eme,%
880         Douzi\`eme,Treizi\`eme,Quatorzi\`eme,Quinzi\`eme,Seizi\`eme,%
```

“Première partie” instead of “Part I”.

```

881      Dix-septi\‘eme,Dix-huiti\‘eme,Dix-neuvi\‘eme,Vingti\‘eme}
882 \AfterBabelCommands{%
883   \DeclareRobustCommand*\FB@emptypart\{\def\thepart\}%
884   \DeclareRobustCommand*\FB@Rpart\{\def\thepart{\Roman{part}}\}%
885 }
886 \SetString\partname\{%
887   \csname ordinal\romannumeral\value{part}\endcsname\space
888   partie\FB@emptypart}
889 \EndBabelCommands

```

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

\CaptionSeparator Let’s consider now captions in figures and tables. In French, captions in figures and tables should never be printed as ‘Figure 1:’ which is the default in standard 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 L^AT_EX_{2_ε} according to Frank Mittelbach), is saved in `\STD@makecaption`. ‘AtBeginDocument’ we compare it to its current definition (some classes like `memoir`, `koma-script` classes, AMS classes, `ua-thesis.cls`...change it). If they are identical, frenchb just adds a hook called `\FBCaption@Separator` to `\@makecaption`; `\FBCaption@Separator` defaults to ‘:’ as in the standard `\@makecaption` and will be changed to ‘:’ in French ‘AtBeginDocument’; it can be also set to `\CaptionSeparator` (‘–’) using [CustomiseFigTabCaptions](#).

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

```

890 \bgroup
891 \catcode'::=12 \catcode'>:=12 \relax
892 \long\gdef\STD@makecaption#1#2{%
893   \vskip\abovecaptionskip
894   \sbox\@tempboxa{#1: #2}%
895   \ifdim \wd\@tempboxa >\hsize
896     #1: #2\par
897   \else
898     \global \@minipagefalse
899     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
900   \fi
901   \vskip\belowcaptionskip}
902 \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.

```

903 \newif\if@FBwarning@capsep
904 \@FBwarning@capseptrue
905 \newcommand{\FBWarning}[2]{\PackageWarning{#1}{#2}}
906 \newcommand*{\CaptionSeparator}{\space\textendash\space}
907 \def\FBCaption@Separator{:\space}
908 \long\def\FB@makecaption#1#2{%
909   \vskip\abovcaptionskip
910   \sbox\@tempboxa{#1\FBCaption@Separator #2}%
911   \ifdim \wd\@tempboxa >\hsize
912     #1\FBCaption@Separator #2\par
913   \else
914     \global \@minipagefalse
915     \hbxt\@hsize{\hfil\box\@tempboxa\hfil}%
916   \fi
917   \vskip\belowcaptionskip}

```

Disable the standard warning with AMS and SMF classes.

```

918 \@ifclassloaded{amsart}{\@FBwarning@capsepfalse}{}
919 \@ifclassloaded{amsbook}{\@FBwarning@capsepfalse}{}
920 \@ifclassloaded{amstex}{\@FBwarning@capsepfalse}{}
921 \@ifclassloaded{amstex}{\@FBwarning@capsepfalse}{}
922 \@ifclassloaded{amproc}{\@FBwarning@capsepfalse}{}
923 \@ifclassloaded{smfart}{\@FBwarning@capsepfalse}{}
924 \@ifclassloaded{smfbook}{\@FBwarning@capsepfalse}{}

```

Disable the standard warning unless high punctuation is active.

```

925 \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.

```

926 \newif\ifFB@koma
927 \@ifclassloaded{memoir}{\@FBwarning@capsepfalse}{}
928 \@ifclassloaded{scrartcl}{\@FBwarning@capsepfalse\FB@komatrue}{}
929 \@ifclassloaded{scrbook}{\@FBwarning@capsepfalse\FB@komatrue}{}
930 \@ifclassloaded{scrreprt}{\@FBwarning@capsepfalse\FB@komatrue}{}
931 \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.

```

932 \@ifpackageloaded{caption}
933   {\FBWarning{frenchb.ldf}%
934     {Please load the "caption" package\MessageBreak
935       AFTER babel/frenchb; reported}%
936   \@FBwarning@capsepfalse}%
937 {}

```

Same for package floatrow.

```

938 \@ifpackageloaded{floatrow}
939   {\FBWarning{frenchb.ldf}%
940     {Please load the "floatrow" package\MessageBreak
941       AFTER babel/frenchb; reported}%
942     \@FBwarning@capsepfalse}%
943   {}

```

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’).

```

944 \AtBeginDocument{%
945   \ifx\@makecaption\STD@makecaption
946     \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.

```

947   \ifFBoldFigTabCaptions
948   \else
949     \def\FBCaption@Separator{\iflanguage{french}{ : }{: }}%
950   \fi
951   \ifFBCustomiseFigTabCaptions
952     \ifx\bbl@main@language\FB@french
953       \def\FBCaption@Separator{\CaptionSeparator}%
954     \fi
955   \fi
956   \@FBwarning@capsepfalse
957 \fi
958 \if@FBwarning@capsep
959   \FBWarning{frenchb.ldf}%
960   {Figures' and tables' captions might look like\MessageBreak
961     'Figure 1:' which is wrong in French.\MessageBreak
962     Check your class or packages to change this;\MessageBreak
963     reported}%
964 \fi
965 \let\FB@makecaption\relax
966 \let\STD@makecaption\relax
967 }

```

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.

```

968 \ifFBunicode
969   \let\FBtextellipsis\textellipsis
970 \else
971   \DeclareTextSymbol{\FBtextellipsis}{LY1}{133}
972   \DeclareTextCommandDefault{\FBtextellipsis}{%
973     .\kern\fontdimen3\font.\kern\fontdimen3\font.\xspace}
974 \fi

```

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

```

975 \newcommand*{\Tdots@}{\@xp\textellipsis}
976 \newcommand*{\Mdots@}{\@xp\mdots@}
977 \AtBeginDocument{\DeclareRobustCommand*{\dots}{\relax
978                   \csname\ifmode M\else T\fi dots@\endcsname}%
979                   \ifdefined\@xp\else\let\@xp\relax\fi
980                   \ifdefined\mdots@\else\let\Mdots@\mathellipsis\fi
981                   }
982 \def\bbl@frenchdots{\babel@save\Tdots@ \let\Tdots@\FBtextellipsis}
983 \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.

```

984 \newcommand*{\frenchbsetup}[1]{%
985   \setkeys{FB}{#1}%
986 }%
987 \@onlypreamble\frenchbsetup

```

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

```

988 \newif\ifFBStandardLayout      \FBStandardLayouttrue
989 \newif\ifFBGlobalLayoutFrench  \FBGlobalLayoutFrenchtrue
990 \newif\ifFBReduceListSpacing   \FBReduceListSpacingfalse
991 \newif\ifFBListOldLayout        \FBListOldLayoutfalse
992 \newif\ifFBCompactItemize       \FBCompactItemizefalse
993 \newif\ifFBStandardItemizeEnv   \FBStandardItemizeEnvtrue
994 \newif\ifFBStandardEnumerateEnv \FBStandardEnumerateEnvtrue
995 \newif\ifFBStandardItemLabels   \FBStandardItemLabelstrue
996 \newif\ifFBStandardLists        \FBStandardListstrue
997 \newif\ifFBIndentFirst          \FBIndentFirstfalse
998 \newif\ifFBFrenchFootnotes      \FBFrenchFootnotesfalse
999 \newif\ifFBAutoSpaceFootnotes   \FBAutoSpaceFootnotesfalse
1000 \newif\ifFBOriginalTypewriter   \FBOriginalTypewriterfalse
1001 \newif\ifFBThinColonSpace       \FBThinColonSpacefalse
1002 \newif\ifFBINGuillSpace         \FBINGuillSpacefalse
1003 \newif\ifFBThinSpaceInFrenchNumbers \FBThinSpaceInFrenchNumbersfalse
1004 \newif\ifFBFrenchSuperscripts   \FBFrenchSuperscriptstrue
1005 \newif\ifFBLowercaseSuperscripts \FBLowercaseSuperscriptstrue
1006 \newif\ifFBPartNameFull          \FBPartNameFulltrue
1007 \newif\ifFBCustomiseFigTabCaptions \FBCustomiseFigTabCaptionsfalse
1008 \newif\ifFBOldFigTabCaptions    \FBOldFigTabCaptionsfalse
1009 \newif\ifFBSuppressWarning       \FBSuppressWarningfalse
1010 \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{}`. When the beamer is loaded, lists are not customised at all to ensure compatibility.

```

1011 \edef\FB@french{\CurrentOption}
1012 \AtEndOfPackage{%
1013   \ifx\bbl@main@language\FB@french
1014     \FBGlobalLayoutFrenchtrue
1015     \@ifclassloaded{beamer}%
1016       {\PackageInfo{frenchb.ldf}{%
1017         No list customisation for the beamer class,%
1018         \MessageBreak reported}}%
1019       {\FBReduceListSpacingtrue
1020        \FBStandardItemizeEnvfalse
1021        \FBStandardEnumerateEnvfalse
1022        \FBStandardItemLabelsfalse}%
1023     \FBIndentFirsttrue
1024     \FBFrenchFootnotesttrue
1025     \FBAutoSpaceFootnotesttrue
1026     \FBCustomiseFigTabCaptionstrue
1027   \else

```

```

1028 \FBGlobalLayoutFrenchfalse
1029 \fi

```

frenchb being an option of babel, it cannot load a package (keyval) while frenchb.ldb is read, so we defer the loading of keyval and the options setup at the end of babel's loading.

```

1030 \RequirePackage{keyval}%
1031 \define@key{FB}{StandardLayout}[true]%
1032     {\csname FBStandardLayout#1\endcsname
1033     \ifFBStandardLayout
1034         \FBReduceListSpacingfalse
1035         \FBStandardItemizeEnvtrue
1036         \FBStandardItemLabelstrue
1037         \FBStandardEnumerateEnvtrue
1038         \FBIndentFirstfalse
1039         \FBFrenchFootnotesfalse
1040         \FBAutoSpaceFootnotesfalse
1041         \FBGlobalLayoutFrenchfalse
1042     \else
1043         \FBReduceListSpacingtrue
1044         \FBStandardItemizeEnvfalse
1045         \FBStandardItemLabelsfalse
1046         \FBStandardEnumerateEnvfalse
1047         \FBIndentFirsttrue
1048         \FBFrenchFootnotesttrue
1049         \FBAutoSpaceFootnotesttrue
1050     \fi}%
1051 \define@key{FB}{GlobalLayoutFrench}[true]%
1052     {\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.

```

1053     \ifFBGlobalLayoutFrench
1054         \ifx\bbbl@main@language\FB@french
1055         \else
1056             \PackageWarning{frenchb.ldb}%
1057                 {Option 'GlobalLayoutFrench' skipped:%
1058                 \MessageBreak French is *not*
1059                 babel's last option.\MessageBreak}%
1060         \fi
1061     \fi}%
1062 \define@key{FB}{ReduceListSpacing}[true]%
1063     {\csname FBReduceListSpacing#1\endcsname}%
1064 \define@key{FB}{ListOldLayout}[true]%
1065     {\csname FBListOldLayout#1\endcsname
1066     \ifFBListOldLayout
1067         \FBStandardEnumerateEnvtrue
1068         \renewcommand*{\FrenchLabelItem}{\textendash}%
1069     \fi}%
1070 \define@key{FB}{CompactItemize}[true]%

```

```

1071         {\csname FBCompactItemize#1\endcsname
1072         \ifFBCompactItemize
1073             \FBStandardItemizeEnvfalse
1074             \FBStandardEnumerateEnvfalse
1075         \else
1076             \FBStandardItemizeEnvtrue
1077             \FBStandardEnumerateEnvtrue
1078         \fi}%
1079 \define@key{FB}{StandardItemizeEnv}[true]%
1080     {\csname FBStandardItemizeEnv#1\endcsname}%
1081 \define@key{FB}{StandardEnumerateEnv}[true]%
1082     {\csname FBStandardEnumerateEnv#1\endcsname}%
1083 \define@key{FB}{StandardItemLabels}[true]%
1084     {\csname FBStandardItemLabels#1\endcsname}%
1085 \define@key{FB}{ItemLabels}{%
1086     \renewcommand*{\FrenchLabelItem}{#1}}%
1087 \define@key{FB}{ItemLabeli}{%
1088     \renewcommand*{\Frlabelitemi}{#1}}%
1089 \define@key{FB}{ItemLabelii}{%
1090     \renewcommand*{\Frlabelitemii}{#1}}%
1091 \define@key{FB}{ItemLabeliii}{%
1092     \renewcommand*{\Frlabelitemiii}{#1}}%
1093 \define@key{FB}{ItemLabeliv}{%
1094     \renewcommand*{\Frlabelitemiv}{#1}}%
1095 \define@key{FB}{StandardLists}[true]%
1096     {\csname FBStandardLists#1\endcsname
1097     \ifFBStandardLists
1098         \FBReduceListSpacingfalse
1099         \FBCompactItemizefalse
1100         \FBStandardItemizeEnvtrue
1101         \FBStandardEnumerateEnvtrue
1102         \FBStandardItemLabelstrue
1103     \else
1104         \FBReduceListSpacingtrue
1105         \FBCompactItemizetrue
1106         \FBStandardItemizeEnvfalse
1107         \FBStandardEnumerateEnvfalse
1108         \FBStandardItemLabelsfalse
1109     \fi}%
1110 \define@key{FB}{IndentFirst}[true]%
1111     {\csname FBIndentFirst#1\endcsname}%
1112 \define@key{FB}{FrenchFootnotes}[true]%
1113     {\csname FBFrenchFootnotes#1\endcsname}%
1114 \define@key{FB}{AutoSpaceFootnotes}[true]%
1115     {\csname FBAutoSpaceFootnotes#1\endcsname}%
1116 \define@key{FB}{AutoSpacePunctuation}[true]%
1117     {\csname FBAutoSpacePunctuation#1\endcsname}%
1118 \define@key{FB}{OriginalTypewriter}[true]%
1119     {\csname FBOriginalTypewriter#1\endcsname}%
1120 \define@key{FB}{ThinColonSpace}[true]%
1121     {\csname FBThinColonSpace#1\endcsname}%

```

```

1122 \define@key{FB}{INGuillSpace}[true]%
1123         {\csname FBINGuillSpace#1\endcsname}%
1124 \define@key{FB}{ThinSpaceInFrenchNumbers}[true]%
1125         {\csname FBThinSpaceInFrenchNumbers#1\endcsname}%
1126 \define@key{FB}{FrenchSuperscripts}[true]%
1127         {\csname FBFrenchSuperscripts#1\endcsname}%
1128 \define@key{FB}{LowercaseSuperscripts}[true]%
1129         {\csname FBLowercaseSuperscripts#1\endcsname}%
1130 \define@key{FB}{PartNameFull}[true]%
1131         {\csname FBPartNameFull#1\endcsname
1132         \ifFBPartNameFull
1133         \else
1134         \FB@addto{captions}{%
1135         \def\partname{Partie\protect\FB@Rpart}}}%
1136         \fi}%
1137 \define@key{FB}{CustomiseFigTabCaptions}[true]%
1138         {\csname FBCustomiseFigTabCaptions#1\endcsname}%
1139 \define@key{FB}{OldFigTabCaptions}[true]%
1140         {\csname FBOldFigTabCaptions#1\endcsname}
    \CurrentOption no longer defined. It's value has been saved in \FB@CurOpt while
    reading frenchb.ldf.
1141         \ifFBOldFigTabCaptions
1142         \FB@addto{extras}{\babel@save\FBCaption@Separator
1143         \def\FBCaption@Separator{\CaptionSeparator}}}%
1144         \fi}%
1145 \define@key{FB}{SuppressWarning}[true]%
1146         {\csname FBSuppressWarning#1\endcsname
1147         \ifFBSuppressWarning
1148         \renewcommand{\FBWarning}[2]{\relax}%
1149         \fi}%
1150 \define@key{FB}{ShowOptions}[true]%
1151         {\csname FBShowOptions#1\endcsname}%
    Inputting French quotes as single characters when they are available on the key-
    board (through a compose key for instance) is more comfortable than typing \og
    and \fg.
    With pdfTeX (or old LuaTeX and XeTeX engines), quote characters are made active
    and expand to \og\ignorespaces and {\fg} respectively if the current language is
    French, and to \guillemotleft and \guillemotright otherwise (think of German
    quotes), this is done by \FB@@og and \FB@@fg; thus correct unbreakable spaces will
    be added automatically to French quotes. The quote characters typed in depend on
    the input encoding, it can be single-byte (latin1, latin9, applemac,...) or multi-bytes
    (utf-8, utf8x); the inputenc package has to be loaded before the \begin{document}
    with the proper coding option, so we check if \DeclareInputText is defined.
    Life is much simpler here with modern LuaTeX or XeTeX engines: we just have
    to activate the \FB@addGUILLspace attribute for LuaTeX or set \XeTeXcharclass of
    quotes to the proper value for XeTeX.
1152 \define@key{FB}{og}{%
1153         \ifFB@active@punct

```

```

1154         \newcommand*{\FB@@og}{%
1155             \iflanguage{french}%
1156                 {\ifFBAutoSpaceGuill\FB@og\ignorespaces
1157                 \else\guillemotleft
1158                 \fi}%
1159             {\guillemotleft}}%
1160     \fi
1161     \ifFBunicode

```

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

```

1162         \ifFB@luatex@punct
1163             \FB@addGUILspace=1 \relax
1164         \else

```

then modern XeTeX, this more tricky:

```

1165         \ifFB@xetex@punct
        \XeTeXinterchartokenstate is defined, we just need to set \XeTeXcharclass to
        \FB@guilo for the French opening quote (see subsection 2.2).
1166         \XeTeXcharclass"AB = \FB@guilo
1167         \XeTeXcharclass"A0 = \FB@guilnul
1168         \XeTeXcharclass"202F = \FB@guilnul
1169     \else

```

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

```

1170         \catcode'#1=\active
1171         \bgroup
1172         \uccode'\~'#1%
1173         \uppercase{%
1174         \egroup
1175         \def~%
1176         }{\FB@@og}%
1177     \fi
1178     \fi
1179     \else

```

This is for conventional TeX engines:

```

1180         \AtBeginDocument{%
1181             \ifdefined\DeclareInputText
1182                 \ifdefined\uc@dcl

```

Package `inputenc` with `utf8x` encoding loaded, use `\uc@dcl`,

```

1183             \uc@dcl{171}{default}{\FB@@og}%
1184         \else

```

if encoding is not `utf8x`, try `utf8`...

```

1185             \ifdefined\DeclareUnicodeCharacter
        utf8 loaded, use \DeclareUnicodeCharacter,
1186             \DeclareUnicodeCharacter{00AB}{\FB@@og}%
1187         \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.

```

1188             \@tempcnta'#1\relax
1189             \ifdefined\mule@def
1190                 \mule@def{11}{\FB@@og}%
1191             \else
1192                 \DeclareInputText{\the\@tempcnta}{\FB@@og}%
1193             \fi
1194         \fi
1195     \fi
1196 \else
    Package inputenc not loaded, no way...
1197     \PackageWarning{frenchb.ldb}{%
1198         {Option 'og' requires package inputenc.\MessageBreak}%
1199     \fi
1200 }%
1201 \fi
1202 }%

```

Same code for the closing quote.

```

1203 \define@key{FB}{fg}{%
1204     \ifFB@active@punct
1205         \newcommand*{\FB@@fg}{%
1206             \iflanguage{french}%
1207                 {\ifFBAutoSpaceGuill\FB@fg
1208                 \else\guillemotright
1209                 \fi}%
1210             {\guillemotright}}%
1211     \fi
1212     \ifFBunicode
1213         \ifFB@luatex@punct
1214             \FB@addGUILspace=1 \relax
1215         \else
1216             \ifFB@xetex@punct
1217                 \XeTeXcharclass"BB = \FB@guilf
1218                 \XeTeXcharclass"A0 = \FB@guilnul
1219                 \XeTeXcharclass"202F = \FB@guilnul
1220             \else
1221                 \catcode'#1=\active
1222                 \bgroup
1223                 \uccode'\~'#1%
1224                 \uppercase{%
1225                     \egroup
1226                     \def~%
1227                     }{\FB@@fg}}%
1228             \fi
1229         \fi
1230     \else
1231         \AtBeginDocument{%
1232             \ifdefined\DeclareInputText
1233                 \ifdefined\uc@dc1c

```

```

1234         \uc@dc{187}{default}}{\FB@fg}%
1235     \else
1236         \ifdefined\DeclareUnicodeCharacter
1237         \DeclareUnicodeCharacter{00BB}{\FB@fg}%
1238     \else
1239         \@tempcnta'#1\relax
1240         \ifdefined\mule@def
1241         \mule@def{27}{\FB@fg}%
1242     \else
1243         \DeclareInputText{\the\@tempcnta}{\FB@fg}%
1244     \fi
1245 \fi
1246 \fi
1247 \else
1248     \PackageWarning{frenchb.ldb}%
1249     {Option 'fg' requires package inputenc.\MessageBreak}%
1250 \fi
1251 }%
1252 \fi
1253 }%
1254 }

```

\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.

```

1255 \newcommand*{\FBprocess@options}{%

```

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

```

1256 \ifpackageloaded{enumitem}{%
1257     \ifFBStandardItemizeEnv
1258     \else
1259         \FBStandardItemizeEnvtrue
1260         \PackageInfo{frenchb.ldb}%
1261         {Setting StandardItemizeEnv=true for\MessageBreak
1262         compatibility with enumitem package,\MessageBreak}%
1263     \fi
1264     \ifFBStandardEnumerateEnv
1265     \else
1266         \FBStandardEnumerateEnvtrue
1267         \PackageInfo{frenchb.ldb}%
1268         {Setting StandardEnumerateEnv=true for\MessageBreak
1269         compatibility with enumitem package,\MessageBreak}%
1270     \fi}}%
1271 \ifpackageloaded{paralist}{%
1272     \ifFBStandardItemizeEnv
1273     \else

```



```

1274     \FBStandardItemizeEnvtrue
1275     \PackageInfo{frenchb.ldb}%
1276     {Setting StandardItemizeEnv=true for\MessageBreak
1277      compatibility with paralist package,\MessageBreak}%
1278   \fi
1279   \ifFBStandardEnumerateEnv
1280   \else
1281     \FBStandardEnumerateEnvtrue
1282     \PackageInfo{frenchb.ldb}%
1283     {Setting StandardEnumerateEnv=true for\MessageBreak
1284      compatibility with paralist package,\MessageBreak}%
1285   \fi}{}%
1286 \@ifpackageloaded{enumerate}{%
1287   \ifFBStandardEnumerateEnv
1288   \else
1289     \FBStandardEnumerateEnvtrue
1290     \PackageInfo{frenchb.ldb}%
1291     {Setting StandardEnumerateEnv=true for\MessageBreak
1292      compatibility with enumerate package,\MessageBreak}%
1293   \fi}{}%

```

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

```

1294 \def\FB@ufl{\update@frenchlists}
1295 \ifx\bbbl@main@language\FB@french
1296   \update@frenchlists
1297 \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.

```

1298 \ifFBAutoSpacePunctuation
1299   \autospace@beforeFDP
1300 \else
1301   \noautospace@beforeFDP
1302 \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.

```

1303 \ifFBOriginalTypewriter
1304 \else
1305   \let\ttfamilyORI\ttfamily
1306   \let\rmfamilyORI\rmfamily
1307   \let\sffamilyORI\sffamily
1308   \let\ttfamily\ttfamilyFB
1309   \let\rmfamily\rmfamilyFB
1310   \let\sffamily\sffamilyFB
1311 \fi

```

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

```

1312 \ifFBThinColonSpace
1313   \ifFB@luatex@punct
1314     \FBcolonskip=\FBthinskip\relax
1315   \else
1316     \renewcommand*{\FBcolonspace}{\FBthinspace}%
1317   \fi
1318 \fi

```

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

```

1319 \ifFBINGuillSpace
1320   \ifFB@luatex@punct
1321     \FBguillskip=3.33pt plus 1.665pt minus 1.11pt \relax
1322   \else
1323     \renewcommand*{\FBguillspace}{\space}%
1324   \fi
1325 \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.

```

1326 \@ifpackageloaded{numprint}%
1327 {\ifnprt@autolanguage
1328   \providecommand*{\npstylefrench}{}%
1329   \ifFBThinSpaceInFrenchNumbers
1330     \renewcommand*\npstylefrench{%
1331       \npthousandsep{\,}%
1332       \npdecimalsign{,}%
1333       \npproductsign{\cdot}%
1334       \npunitseparator{\,}%
1335       \npdegreeseperator{}%
1336       \nppercentseparator{\nprt@unitsep}%
1337     }%
1338   \else
1339     \renewcommand*\npstylefrench{%
1340       \npthousandsep{~}%
1341       \npdecimalsign{,}%
1342       \npproductsign{\cdot}%
1343       \npunitseparator{\,}%
1344       \npdegreeseperator{}%
1345       \nppercentseparator{\nprt@unitsep}%
1346     }%
1347   \fi
1348   \npaddtolanguage{french}{french}%
1349 \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.

```

1350 \ifFBFrenchSuperscripts
1351   \DeclareRobustCommand*{\up}{\@ifstar{\FB@up@fake}{\fup}}%
1352 \else
1353   \DeclareRobustCommand*{\up}{\@ifstar{\FB@up@fake}%
1354                                     {\textsuperscript}}%
1355 \fi

```

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

```

1356 \ifFBLowercaseSuperscripts
1357 \else
1358   \renewcommand*{\FB@lc}[1]{##1}%
1359 \fi

```

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

```

1360 \ifBFCustomiseFigTabCaptions
1361   \ifFB@koma
1362     \let\captionformat\CaptionSeparator
1363   \fi
1364   \@ifclassloaded{memoir}%
1365     {\captiondelim{\CaptionSeparator}}{}%
1366 \else

```

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

```

1367   \ifFB@koma
1368     \def\captionformat{\iflanguage{french}{ : }{: }}%
1369   \fi
1370   \@ifclassloaded{memoir}%
1371     {\captiondelim{\iflanguage{french}{ : }{: }}}{}%
1372 \fi

```

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

```

1373 \ifFBShowOptions
1374   \GenericWarning{* }{%
1375     * **** List of possible options for frenchb ****\MessageBreak
1376     [Default values between brackets when frenchb is loaded *LAST*]%
1377     \MessageBreak
1378     ShowOptions=true [false]\MessageBreak
1379     StandardLayout=true [false]\MessageBreak
1380     GlobalLayoutFrench=false [true]\MessageBreak
1381     StandardLists=true [false]\MessageBreak
1382     IndentFirst=false [true]\MessageBreak
1383     ReduceListSpacing=false [true]\MessageBreak
1384     ListOldLayout=true [false]\MessageBreak
1385     StandardItemizeEnv=true [false]\MessageBreak
1386     StandardEnumerateEnv=true [false]\MessageBreak
1387     StandardItemLabels=true [false]\MessageBreak
1388     ItemLabels=\textemdash, \textbullet,

```

```

1389      \protect\ding{43},... [\textendash]\MessageBreak
1390      ItemLabeli=\textendash, \textbullet,
1391      \protect\ding{43},... [\textendash]\MessageBreak
1392      ItemLabelii=\textendash, \textbullet,
1393      \protect\ding{43},... [\textendash]\MessageBreak
1394      ItemLabeliii=\textendash, \textbullet,
1395      \protect\ding{43},... [\textendash]\MessageBreak
1396      ItemLabeliv=\textendash, \textbullet,
1397      \protect\ding{43},... [\textendash]\MessageBreak
1398      FrenchFootnotes=false [true]\MessageBreak
1399      AutoSpaceFootnotes=false [true]\MessageBreak
1400      AutoSpacePunctuation=false [true]\MessageBreak
1401      OriginalTypewriter=true [false]\MessageBreak
1402      ThinColonSpace=true [false]\MessageBreak
1403      INGuillSpace=true [false]\MessageBreak
1404      ThinSpaceInFrenchNumbers=true [false]\MessageBreak
1405      FrenchSuperscripts=false [true]\MessageBreak
1406      LowercaseSuperscripts=false [true]\MessageBreak
1407      PartNameFull=false [true]\MessageBreak
1408      SuppressWarning=true [false]\MessageBreak
1409      CustomiseFigTabCaptions=false [true]\MessageBreak
1410      OldFigTabCaptions=true [false]\MessageBreak
1411      og= <left quote character>, fg= <right quote character>%
1412      \MessageBreak
1413      %*****%
1414      \MessageBreak\protect\frenchbsetup{ShowOptions}}
1415  \fi
1416 }

```

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.

```

1417 \AtBeginDocument{%
1418   \providecommand*\xspace{\relax}%

```

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

```

1419   \ifdefined\pdfstringdefDisableCommands
1420     \pdfstringdefDisableCommands{%
1421       \let\up\relax
1422       \let\up\relax
1423       \let\degre\textdegree
1424       \let\degres\textdegree
1425       \def\ieme{e\xspace}%
1426       \def\iemes{es\xspace}%
1427       \def\ier{er\xspace}%
1428       \def\iers{ers\xspace}%
1429       \def\iere{re\xspace}%
1430       \def\ieres{res\xspace}%
1431       \def\FrenchEnumerate#1{#1\degre\space}%
1432       \def\FrenchPopularEnumerate#1{#1\degre)\space}%

```

```

1433      \def\No{N\degre\space}%
1434      \def\no{n\degre\space}%
1435      \def\Nos{N\degre\space}%
1436      \def\nos{n\degre\space}%
1437      \def\FB@og{\guillemotleft\space}%
1438      \def\FB@fg{\space\guillemotright}%
1439      \def\at{@}%
1440      \def\circonflexe{\string^}%
1441      \def\tild{\string~}%
1442      \let\bsc\textsc
1443    }%
1444  \fi

```

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

```

1445  \FBprocess@options

```

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

```

1446  \ifFB@luatex@punct
1447    \activate@luatexpunct
1448  \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!

```

1449  \ifFBunicode
1450    \ifdefined\DeclareUTFcharacter
1451    \else
1452      \PackageWarning{frenchb.ldf}%
1453        {Add \protect\usepackage{fontspec} to the\MessageBreak
1454        preamble of your document,}%
1455    \fi
1456  \else
1457    \begingroup \newcommand{\FBOTone}{OT1}%
1458    \ifx\encodingdefault\FBOTone
1459      \PackageWarning{frenchb.ldf}%
1460        {OT1 encoding should not be used for French.%
1461        \MessageBreak
1462        Add \protect\usepackage[T1]{fontenc} to the
1463        preamble\MessageBreak of your document,}%
1464    \fi
1465  \endgroup
1466  \fi
1467 }

```

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 `\FB@listVsettings`

when entering French and restoring them to their defaults on exit would not work; so we define the command `\FB@listVsettings` to hold the settings to be used by the French variant `\listFB` of `\list`. Note that switching to `\listFB` reduces vertical spacing in *all* environments built on `\list`: `itemize`, `enumerate`, `description`, but also `abstract`, `quotation`, `quote` and `verse`...

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

```
1468 \let\listORI\list
1469 \let\endlistORI\endlist
1470 \def\FB@listVsettings{%
1471     \setlength{\itemsep}{0.4ex plus 0.2ex minus 0.2ex}%
1472     \setlength{\parsep}{0.4ex plus 0.2ex minus 0.2ex}%
1473     \setlength{\topsep}{0.8ex plus 0.4ex minus 0.4ex}%
1474     \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`.

```
1475     \@tempdima=\parskip
1476     \addtolength{\topsep}{-\@tempdima}%
1477     \addtolength{\partopsep}{\@tempdima}%
1478 }
1479 \def\listFB#1#2{\listORI{#1}{\FB@listVsettings #2}}
1480 \let\endlistFB\endlist
```

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

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

<p>Text starting at 'parindent'</p> <p>⇐ Leftmargin</p> <p>– first item. . .</p> <p>– first second level item</p> <p>– next one. . .</p> <p>– second item. . .</p>
--

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

```
\Frlabelitemi1481 \newcommand*{\FrenchLabelItem}{\textemdash}
\Frlabelitemii1482 \newcommand*{\Frlabelitemi}{\FrenchLabelItem}
\Frlabelitemiii
\Frlabelitemiv
```

```

1483 \newcommand*{\Frlabelitemii}{\FrenchLabelItem}
1484 \newcommand*{\Frlabelitemiii}{\FrenchLabelItem}
1485 \newcommand*{\Frlabelitemiv}{\FrenchLabelItem}

```

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

```

1486 \newlength\listindentFB
1487 \setlength{\listindentFB}{-1pt}
1488 \newlength\labelwidthFB
1489 \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.

```

1490 \newlength\leftmarginFB
1491 \def\FB@listHsettings{%
1492   \leftmarginFB\labelwidthFB
1493   \advance\leftmarginFB \labelsep
1494   \leftmarginii\leftmarginFB
1495   \advance\leftmarginii \listindentFB
1496   \leftmarginiii\leftmarginFB
1497   \leftmarginiiii\leftmarginFB
1498   \leftmarginiv\leftmarginFB
1499   \leftmargin\cename leftmargin\romannumeral\the\@listdepth\endcsname
1500 }

```

\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).

```

1501 \def\FB@itemizesettings{%
1502   \ifFBReduceListSpacing
1503     \setlength{\itemsep}{\z@}%
1504     \setlength{\parsep}{\z@}%
1505     \setlength{\topsep}{\z@}%
1506     \setlength{\partopsep}{\z@}%
1507     \@tempdima=\parskip
1508     \addtolength{\topsep}{-\@tempdima}%
1509     \addtolength{\partopsep}{-\@tempdima}%
1510   \fi
1511   \settowidth{\labelwidth}{\cename\@itemitem\endcsname}%
1512   \ifBListOldLayout
1513     \setlength{\leftmargin}{\labelwidth}%
1514     \addtolength{\leftmargin}{\labelsep}%
1515     \addtolength{\leftmargin}{\parindent}%
1516   \else

```

```

1517     \FB@listHsettings
1518     \fi
1519 }

```

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`.

```

1520 \def\itemizeFB{%
1521     \ifnum \@itemdepth >\thr@@\@toodeep\else
1522         \advance\@itemdepth\@ne
1523         \edef\@itemitem{labelitem\romannumeral\the\@itemdepth}%
1524         \expandafter
1525         \listORI
1526         \csname\@itemitem\endcsname
1527         \FB@itemizesettings
1528     \fi
1529 }
1530 \let\enditemizeFB\endlistORI

1531 \def\labelitemFB{%
1532     \let\labelitemi\Frlabelitemi
1533     \let\labelitemii\Frlabelitemii
1534     \let\labelitemiii\Frlabelitemiii
1535     \let\labelitemiv\Frlabelitemiv
1536     \ifdim\labelwidthFB<\z@
1537         \settowidth{\labelwidthFB}{\FrenchLabelItem}%
1538     \fi
1539     \ifdim\listindentFB<\z@
1540         \ifdim\parindent=\z@
1541             \setlength{\listindentFB}{1.5em}%
1542         \else
1543             \setlength{\listindentFB}{\parindent}%
1544         \fi
1545     \fi
1546 }

```

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

```

1547 \def\enumerateFB{%
1548     \ifnum \@enumdepth >\thr@@\@toodeep\else
1549         \advance\@enumdepth\@ne
1550         \edef\@enumctr{enum\romannumeral\the\@enumdepth}%
1551         \expandafter
1552         \list
1553         \csname label\@enumctr\endcsname
1554         {\FB@listHsettings
1555         \usecounter\@enumctr\def\makelabel##1{\hss\llap{##1}}}%
1556     \fi
1557 }
1558 \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).

```

1559 \def\descriptionFB{%
1560     \list{}\FB@listHsettings
1561         \labelwidth\z@
1562         \itemindent-\leftmargin
1563         \ifnum\@listdepth=1
1564             \advance\itemindent by \listindentFB
1565         \fi
1566         \let\makelabel\descriptionlabel}%
1567 }
1568 \let\enddescriptionFB\endlistORI

```

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

```

\bbbl@frenchlistlayout 1569 \def\update@frenchlists{%
\bbbl@nonfrenchlistlayout 1570 \ifFBReduceListSpacing \let\list\listFB \fi
1571 \ifFBStandardItemizeEnv
1572 \else \let\itemize\itemizeFB \fi
1573 \ifFBStandardItemLabels
1574 \else \labelitemsFB \fi
1575 \ifFBStandardEnumerateEnv
1576 \else \let\enumerate\enumerateFB \let\description\descriptionFB \fi
1577 }

```

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. 49.

```

1578 \def\FB@ufl{\relax}
1579 \def\bbbl@frenchlistlayout{%
1580     \ifFBGlobalLayoutFrench
1581     \else
1582         \babel@save\list           \babel@save\itemize
1583         \babel@save\enumerate     \babel@save\description
1584         \babel@save\labelitemi    \babel@save\labelitemii
1585         \babel@save\labelitemiii  \babel@save\labelitemiv
1586     \fi
1587     \FB@ufl
1588 }
1589 \def\bbbl@nonfrenchlistlayout{%
1590     \ifFBGlobalLayoutFrench
1591         \update@frenchlists
1592     \fi
1593 }
1594 \FB@addto{extras}{\bbbl@frenchlistlayout}
1595 \FB@addto{noextras}{\bbbl@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.

```

1596 \def\bbl@frenchindent{%
1597   \ifFBGlobalLayoutFrench\else\babel@save\@afterindentfalse\fi
1598   \ifFBIndentFirst
1599     \let\@afterindentfalse\@afterindenttrue
1600     \@afterindenttrue
1601   \fi}
1602 \def\bbl@nonfrenchindent{%
1603   \ifFBGlobalLayoutFrench
1604     \ifFBIndentFirst
1605       \@afterindenttrue
1606     \fi
1607   \fi}
1608 \FB@addto{extras}{\bbl@frenchindent}
1609 \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.

```

1610 \AtBeginDocument{\@ifpackageloaded{bigfoot}%
1611   {\PackageInfo{frenchb.ldf}%
1612     {bigfoot package in use.\MessageBreak
1613       frenchb will NOT customise footnotes;\MessageBreak
1614       reported}}%
1615   {\let\@footnotemarkORI\@footnotemark
1616     \def\@footnotemarkFB{\leavevmode\unskip\unkern
1617       \,\@footnotemarkORI}%
1618     \ifFBAutoSpaceFootnotes
1619       \let\@footnotemark\@footnotemarkFB
1620     \fi}%
1621   }

```

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).

```

1622 \newcommand*{\dotFFN}{.}
1623 \newcommand*{\kernFFN}{\kern .5em}
1624 \newdimen\parindentFFN
1625 \parindentFFN=10in
1626 \def\ftnISsymbol{\@fnsymbol\c@footnote}
1627 \long\def\@makefntextFB#1{\ifx\thefootnote\ftnISsymbol
1628     \@makefntextORI{#1}%
1629     \else
1630     \parindent=\parindentFFN
1631     \rule{z@}{\footnotesep}
1632     \setbox\@tempboxa\hbox{\@thefnmark}%
1633     \ifdim\wd\@tempboxa>z@
1634     \llap{\@thefnmark}\dotFFN\kernFFN
1635     \fi #1
1636     \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).

```

1637 \AtBeginDocument{\@ifpackageloaded{bigfoot}{}%
1638     {\ifdim\parindentFFN<10in
1639     \else
1640     \parindentFFN=\parindent
1641     \ifdim\parindentFFN<1.5em \parindentFFN=1.5em \fi
1642     \fi
1643     \let\@makefntextORI\@makefntext
1644     \long\def\@makefntext#1{%
1645     \ifFBFrenchFootnotes
1646     \@makefntextFB{#1}%
1647     \else
1648     \@makefntextORI{#1}%
1649     \fi}%
1650     }%
1651     }

```

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`.

```

1652 \newcommand*{\AddThinSpaceBeforeFootnotes}{\FBAutoSpaceFootnotestrue}
1653 \newcommand*{\FrenchFootnotes}{\FBFrenchFootnotestrue}

```

```
1654 \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.

```
1655 \FBclean@on@exit
```

```
1656 \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>).	59	and <code>StandardLayout</code> → <code>StandardLists</code>	41
	Added warning for OT1 encoding.	52		
	Footnotes are now printed by default ‘à la française’ for the whole document.	58		
	New command <code>\frenchbsetup</code> added for global customisation.	41		
	<code>\bsc</code> : <code>\hbox</code> dropped, replaced by <code>\kern0pt</code>	34		
	<code>\captionsfrench</code> : ‘Fig.’ changed to ‘Figure’ and ‘Tab.’ to ‘Table’.	37		
	<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).	41		
	<code>\nombre</code> : <code>\nombre</code> now requires <code>numprint.sty</code>	36		
v2.0b	General: Footnotes: Just do nothing (except warning) when the <code>bigfoot</code> package is loaded.	58		
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>	37		
	<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>			
	<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.	52		
	<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).	35		
	<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). 59
Avoid warning “\end occurred when <code>\ifx ... incomplete</code> ” with LaTeX-2.09. 12	v2.3c
v2.2a	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
<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	v2.3d
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	<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. 58
When frenchb is babel’s last option, French becomes the document’s main language, so GlobalLayout-French applies. 41	v2.3e
<code>\fup</code> : <code>\newif</code> and <code>\newdimen</code> moved before <code>\ifLaTeXe</code> to avoid an error with plainTeX. 31	General: Execute <code>\AutoSpaceBeforeFDP</code> also in LaTeX to define <code>\FDP@colonspace</code> : needed for tex4ht, pointed out by MPG. . . . 27
v2.3a	v2.4a
General: <code>\NoAutoSpaceBeforeFDP</code> and <code>\AutoSpaceBeforeFDP</code> now set the flag <code>\ifFBAutoSpacePunctuation</code> accordingly (LaTeX only). 27	General: <code>\PackageWarning</code> changed to <code>\FBWarning</code> (when bigfoot package in use). 58
In LaTeX, frenchb no longer adds spaces before ‘high punctuation’ characters in computer code. Suggested by Yannis Haralambous. 27	<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>\frenchbsetup</code> : New option: <code>OriginalTypewriter</code> . Now frenchb switches to <code>\noautospace@beforeFDP</code> when a tt-font is in use. When <code>OriginalTypewriter</code> is set to true, frenchb behaves as in pre-2.3 versions. 41	<code>\frenchbsetup</code> : New option <code>SuppressWarning</code> 41
<code>\fup</code> : <code>\lowercase</code> changed to <code>\MakeLowercase</code> as the former doesn’t work for non ASCII characters in encodings like <code>applemac, utf-8,...</code> 31	<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
v2.3b	v2.4c
General: New commands <code>\dotFFN</code> and <code>\kernFFN</code> for more flexibility	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

<p><code>\newcommand</code> as <code>\up</code> may be defined elsewhere (catalan.ldf). Bug pointed out by Felip Manyé i Ballester. 31</p>	<p><code>\ifFBXeTeX</code>: Added two new ‘if’ <code>\FBXeTeX</code> and <code>\FBLuaTeX</code> as XeTeX and behave differently regarding the status of the French “apostrophe”. 14</p>
<p>v2.5a</p> <p>General: <code>\og</code> and <code>\fg</code> do not print correctly in English when using XeTeX or LuaTeX, fixed by using <code>\textquotedblleft</code> and <code>\textquotedblright</code> defined above. 30</p> <p>New command <code>\NoAutoSpacing</code>, suggested by MPG. 28</p> <p>Punctuation is no longer made active with XeTeX-based engines. 15</p> <p><code>\captionsfrench</code>: <code>\emph</code> deleted in <code>\seename</code> and <code>\alsoname</code> to match what is done for the other languages. Suggested by Marc Baudoin. 37</p> <p><code>\FBthinspace</code>: Define <code>\FBthinspace</code> for those who want to customise the width of the space before ; and co. 16</p> <p><code>\textquoteddblright</code>: Change <code>\guillemotleft</code> and <code>\guillemotright</code> definitions for Unicode and provide definitions for <code>\textquotedblleft</code> and <code>\textquotedblright</code>. Insures correct printing of quotes by <code>\og</code> and <code>\fg</code> in French and outside. 29</p>	<p>v2.5e</p> <p>General: <code>\pdfstringdefDisableCommands</code> should redefine <code>\FB@og</code> and <code>\FB@fg</code> instead of <code>\og</code> and <code>\fg</code> so that it works also when quotes are entered as characters. Reported by Sébastien Gouezel. 52</p>
<p>v2.5b</p> <p>General: Do not use the test <code>\iflanguage{french}</code> to check whether French is the main language or not, as it might be erroneously positive when English is the main language and no hyphenation patterns are available for French. In this case <code>\l@french</code> and <code>\l@english</code> are 0. Pointed out by Günter Milde. 42</p>	<p>v2.5f</p> <p>General: Changed definitions of <code>\at</code>, <code>\circonflexe</code>, <code>\tild</code>, <code>\boi</code> and <code>\degre</code> for Unicode based engines. 34</p> <p><code>\FBtextellipsis</code>: Unicode fonts also provide a ready made character for <code>\textellipsis</code>, let’s just use it (reported by Maxime Chupin, 2011/06/04). 41</p>
<p>v2.5c</p> <p>General: The code meant for XeTeX also works for LuaTeX, we just need to change the test. 45</p>	<p>v2.5g</p> <p>General: Add four <code>\newif</code> to control spacing of quotes (characters and control sequences). 22</p> <p>Redefine <code>\degre</code>, <code>\degres</code> <code>\at</code> <code>\circonflexe</code> and <code>\tild</code> for bookmarks. Add <code>\fup</code> also. 52</p> <p>When <code>\ifFB@xetex@punct</code> is true, ‘og’ and ‘fg’ options now set XeTeXcharclasses of these characters to <code>\FB@guilo</code> and <code>\FB@guilf</code>. Otherwise French quotes behave as normal characters (their XeTeX-charclass is 0). 45</p> <p><code>\FB@xetex@punct@french</code>: XeTeX-charclass(es) for French quotes will be set to <code>\FB@guilo</code> and <code>\FB@guilf</code> by options ‘og’ and ‘fg’ in <code>\frenchbsetup</code>. French quotes should behave as normal characters by default in XeLaTeX as in LaTeX. 23</p>
<p>v2.5d</p> <p>General: Moved the <code>\newcount</code> command outside <code>\ifFB@xetex@punct</code> ... <code>\fi</code> (it broke Plain formats). . 22</p>	<p>v2.5h</p> <p><code>\degres</code>: <code>textcomp.sty</code> has changed. The test about <code>\M@TS1</code> is no longer relevant, let’s change it. 35</p>
	<p>v2.5i</p> <p>General: Temporary fix: as long as <code>xeCJK.sty</code> will not use</p>

\newXeTeXintercharclass to allocate its classes, we will have to define 3 fake classes.	23	\CaptionSeparator: Former \CaptionSeparator has been renamed as \FBCaption@Separator; Newif \ifFBwarning@capsep added.	38
\FB@xetex@punct@french: xeCJK.sty changes the \XeTeXcharclass of ASCII chars ‘ ’ ’ ’ ’ ’ ’ ’ ‘ ’ ’ ’ opening and closing single and double quotes. We set their class to 0 in French and reset their class to their original value when leaving French. See \FB@xetex@punct@nonfrench below.	23	v2.6d \FBthinspace: Rename \Fthinspace to \FBthinspace and \Fcolonspace to \FBcolonspace to avoid a conflict with fournier.sty.	16
v2.5j General: Previous fix removed: bug fixed in xeCJK.sty version 3.0.4 (06-May-2012).	23	v2.6e \degres: Refrain from redefining \textdegree from latin1.def, applemac.def, etc. as \degres because it loops in hyperref’s bookmarks. Pointed out by Eddy Flas on fctt.	34
v2.6a General: Bug correction: changing \leftmargin cannot be done only for itemize-lists: it messes up embedded enumerate lists. Pointed out by Denis Bitouzé. Lists have been completely redesigned in frenchb v. 2.6a. An option for backward compatibility is provided.	53	v2.6f \FB@itemizesettings: \labelwidth must be reset, f.i. when an itemize list occurs inside environments based on trivlist which set \labelwidth to 0 (see proof environment in amsthm.sty). Bug pointed out by Julien Hauseux. .	55
\frenchbsetup: New options ListOldLayout, StandardItemizeEnv and StandardEnumerateEnv (CompactItemize is deprecated).	41	v2.6g General: U+00A0 (Unicode nobreakspace) and U+202F (Unicode nobreakthinspace) added to class \FB@punctnul to prevent frenchb from adding it’s own space before ‘high punctuation’ characters.	23
\FrenchLabelItem: defaults to \textendash instead of \textendash up to v. 2.5k.	54	\FB@itemizesettings: Suppress all vertical spaces only if ReduceListSpacing is true. Pointed out by Pierre Willaime.	55
v2.6b \descriptionFB: Settings of \FB@listHsettings should apply to description lists too.	57	\ifFBXeTeX: lccode values for the French “apostrophe” are now the same for XeTeX and LuaTeX. ...	14
v2.6c General: Dummy file frenchb.cfg is no longer generated from frenchb.dtx.	11	v2.6h General: \FG@og and \FG@fg changed: former clumsy code removed. ..	45
No warning about \@makecaption for AMS classes.	39	If \@makecaption is undefined, no warning.	39
No warning about \@makecaption for koma-script classes. \captionformat customised in French.	39	New class \FB@guilnul for characters U+00A0 (Unicode nobreakspace) and U+202F (Unicode nobreakthinspace), to prevent frenchb from adding spurious spaces inside quotes.	23
Warning added when the caption or floatrow package is loaded before babel/frenchb.	39		

\CaptionSeparator: No active cat-	for captionnames.	37
codes in \STD@makecaption's def-	\datefrench: Take advantage of ba-	
inition.	bel's \SetString commands for	
38	\datefrench. Doesn't work with	
v3.0a	Plain (yet?).	30
General: \LdfInit checks	\descriptionFB: Add \listindentFB	
\datefrench instead of	to \itemindent. Suggested by De-	
\captionsfrench to avoid a con-	nis Bitouzé.	57
flict with papertex.cls which loads	\extrasfrench: Take advantage of	
datetime.sty.	babel's \babel@savevariable to	
12	handle apostrophe's \lccode. ..	14
\bbl@nonfrenchguillemets	\FBprocess@options: Changed op-	
deleted, use \babel@save in-	tion ThinColonSpace to make it	
stead.	work also with LuaTeX.	50
30	With koma-script and memoir class,	
Added explicit \FBguillskip for	customise \captionformat and	
LuaTeX.	\captiondelim.	51
29	\FBthinspace: LuaTeX requires	
Definitions of \FB@og and \FB@fg	dimensions: two new skips	
now depend on punctuation han-	\FBcolonskip and \FBthinspace. 16	
dling (LuaTeX / XeTeX / active). 29	\FBthinspace: \FBthinspace is no	
french.cfg will be loaded (if found)	longer a kern but a skip (frenchb	
instead of frenchb.cfg. NO NEED	adds a nobreak penalty before it). 16	
for .cfg files in French anyway. . 60	\frenchbsetup: New options Old-	
In Plain, provide a substi-	FigTabCaptions and Customise-	
tute for \PackageWarning and	FigTabCaptions.	41
\PackageInfo.		
13	v3.0b	
Merging of \captionsfrenchb,	General: frenchb.lua was not found	
\captionsfrancais with	by Lua function dofile (not kpath-	
\captionsfrench deleted in fa-	sea aware). Call function	
vor of new babel 3.9 syntax. ... 38	kpse.find_file first, as suggested	
More informative, less TeXnical	by Paul Gaborit.	21
warning about \@makecaption. . 40	Require luatexbase with LaTeX in	
New flag \ifFB@luatex@punct for	case fontspec has not been loaded	
'high punctuation' management	before babel.	16
with LuaTeX engines.		
15	v3.0c	
New handling of 'high punctuation'	General: Activate option Standard-	
through callbacks with LuaTeX en-	Lists when beamer class is	
gines.	loaded.	42
16	Bug correction: null glues should	
No warning about \@makecaption	not trigger space insertion before	
for SMF classes. No warning ei-	high punctuation. Bug pointed out	
ther with LuaTeX or XeTeX en-	by Benoit Rivet for the 'lstlisting'	
gines.	environment of the listings pack-	
39	age.	19
Options processing completely re-	Changed \FBguill@spacing (inter-	
organised.	nal) to \FBguillspace (public). 29	
41	frenchb requires babel-3.9i. 13	
Support for options frenchb, fran-	Just load luatexbase.sty instead of	
cais, canadien, acadian changed. 12	luaotfload.sty with plain formats. 16	
Test \ifXeTeX changed to		
\ifFBunicode and 'xltextra'		
changed to 'fontspec'.		
53		
\CaptionSeparator: Remove		
\CaptionSeparatorORI, use		
\babel@save instead.		
38		
\captionsfrench: Take advantage		
of babel's \SetString commands		

Missing pair of braces after ‘and’ in the definition of ‘auto’.	19	<code>\datefrench: \SetString</code> still does not work for Plain with babel 3.9k. Need to define <code>\datefrench</code> . . .	30
No need to define <code>\l@french</code> as <code>\lang@french</code> , babel.def (3.9j) takes care for this.	12	<code>\frenchbsetup: New option IN-</code> <code>GuillSpace</code>	41