A Babel language definition file for French

frenchb3.dtx v3.7b, 2025-07-02

Daniel Flipo
daniel.flipo@free.fr

Contents

1 The French language **2**
1.1 Basic interface . . . 2
1.2 Customisation . . . 5
1.2.1 \frenchsetup . . . 5
1.2.2 Caption names . . . 9
1.2.3 Figure and table captions . . . 9
1.3 Hyphenation checks . . . 10
1.4 Changes . . . 12

2 The code **15**
2.1 Initial setup . . . 15
2.2 Punctuation . . . 18
2.2.1 Punctuation with XeTeX . . . 19
2.2.2 Punctuation with standard (pdf)TeX . . . 23
2.2.3 Punctuation switches common to both engines . . . 25
2.3 Commands for French quotation marks . . . 26
2.4 Date in French . . . 31
2.5 Extra utilities . . . 31
2.6 Formatting numbers . . . 35
2.7 Caption names . . . 38
2.8 Figure and table captions . . . 40
2.9 Dots… . . . 42
2.10 More checks about packages' loading order . . . 43
2.11 Setup options: key/value stuff (l3keys) . . . 44
2.12 French lists . . . 58
2.13 French indentation of sections . . . 64
2.14 Formatting footnotes . . . 64
2.15 Clean up and exit . . . 69
2.16 Files `frenchb.ldf`, `francais.ldf`, `canadien.ldf` and `acadian.ldf` . 70

3 Change History **71**

1 The French language

The file `frenchb3.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.
Starting with version 4.0a, the development of `babel-french` is split into to parts: the (legacy) part "frenchb3" meant for TeX pdfTeX and XeTeX which is frozen except for bug corrections and the (active) part "frenchb" meant for LuaTeX.

This file `frenchb3.dtx` is for eTeX, pdfTeX and XeTeX only. See file `frenchb.dtx` for LuaTeX.

`babel-french` has been improved using helpful suggestions from many people, mainly from Jacques André, Michel Bovani, Thierry Bouche, Vincent Jalby, Denis Bitouzé, Thomas Savary, Ulrike Fisher and Marcel Krüger. Thanks to all of them!
LaTeX-2.09 is no longer supported. `babel-french` is designed to be used only with LaTeX2e and Plain formats based on LuaTeX or XeTeX engines.
Changes between version 3.3 (2018) and 3.7 are listed in subsection 1.4 p. 12.
An extensive documentation in French (file `frenchb3-doc.pdf`) is now included in `babel-french`.

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.
The French language can be loaded with `Babel` by a command like:
`\usepackage[german,spanish,french,british]{babel}` [2]
`babel-french` takes account of `Babel`'s *main language* defined as the *last* option at `Babel`'s loading. When French is not `Babel`'s main language, `babel-french` 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 `babel-french`.

[1] The file described in this section has version number 3.7a and was last revised on 2025/05/25.

[2] *Always* use `french` as option name for the French language, former aliases `frenchb` or `francais` are *depreciated*; expect them to be removed sooner or later!

When French is loaded as the last option of `Babel`, `babel-french` makes the following changes to the global layout, *both in French and in all other languages*[3]:

1. the first paragraph of each section is indented (LaTeX 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 `\frenchsetup{}` (see section 1.2 p. 5);
3. vertical spacing in general LaTeX 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 1.2.3 p. 9.

Regarding local typography, the command `\selectlanguage{french}` switches to the French language[4], with the following effects:

1. French hyphenation patterns are made active;
2. 'high punctuation' characters (`: ; ! ?`) automatically add correct spacing [5] in French; this is achieved using '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 (LaTeX only). For customisation of caption names see section 1.2.2 p. 9.
5. the space after `\dots` is removed in French.

Some commands are provided by `babel-french` to make typesetting easier:

1. French quotation marks can be entered using the command `\frquote{}`: `\frquote{some text}` will output « some text ». Former commands `\og` and `\fg` are kept for backward compatibility: `\og some text\fg{}` is an alternative to `\frquote{some text}`.

   If French quote characters are available on your keyboard, you can use them, to get proper spacing in LaTeX2e see option `og=«, fg=»` p. 8.

   For quotations spreading over more than one paragraph, `\frquote` will add at the beginning of every paragraph of the quotation either an opening French guillemet («), or a closing one (») or nothing depending on option `EveryParGuill=open` or `=close` or `=none`, see p. 8.
   The command `\NoEveryParQuote` is provided to locally suppress unwanted

[3] For each item, hooks are provided to reset standard LaTeX settings or to emulate the behavior of former versions of `babel-french` (see command `\frenchsetup{}`, section 1.2 p. 5).

[4] `\selectlanguage{francais}` and `\selectlanguage{frenchb}` are no longer supported.

[5] Well, the automatic insertion may add unwanted spaces in some cases, for correction see `AutoSpacePunctuation` option and `\NoAutoSpacing` command p. 7.

guillemets (typically when lists are embedded in `\frquote{}`), it is meant to be used inside an environment or a group.

`\frquote` is recommended to enter embedded quotations “à la française”, several variants are provided through options. The inner quotation is surrounded by double quotes (“*texte*”) unless option `InnerGuillSingle=true` is selected then a) the inner quotation is printed as ‹ *texte* › and b) if the inner quotation spreads over more than one paragraph, every paragraph included in the inner quotation starts with a ‹ or a › or nothing, depending on option `EveryParGuill=open` (default) or `=close` or `=none`.

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

2. `\frenchdate{`*<year>*`}{`*<month>*`}{`*<day>*`}` helps typesetting dates in French: `\frenchdate{2001}{01}{01}` will print 1er janvier 2001 in a box without any linebreak.

3. A command `\up` is provided to typeset superscripts like `M\up{me}` (abbreviation for “Madame”), `1\up{er}` (for “premier”). Other commands are also provided for ordinals: `\ier`, `\iere`, `\iers`, `\ieres`, `\ieme`, `\iemes` (`3\iemes` prints 3es). All these commands take advantage of real superscript letters when they are available in the current font.

4. Command `\bname{}` (boxed name) is provided to typeset family names: its argument will not be hyphenated except on explicit hyphens. `\bsc{}` (boxed small caps) is a variant that prints its argument in small capitals, it is meant for bibliographies, signatures, etc. Usage: `Albert~\bsc{Camus}`.

5. Commands `\primo`, `\secundo`, `\tertio` and `\quarto` print 1^{o}, 2^{o}, 3^{o}, 4^{o}. `\FrenchEnumerate{6}` prints 6^{o}.

6. Abbreviations for “Numéro(s)” and “numéro(s)” (N^{o} N^{os} n^{o} and n^{os}) are obtained via the commands `\No`, `\Nos`, `\no`, `\nos`.

7. 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 a non-breaking space), or for alcohols” strengths (e.g., “`45\degres`” with *no* space in French) or for angles in math mode.

8. 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 behave as an ordinary character *when the current language is French* (no space added); as a counterpart, if `\DecimalMathComma` is active, an explicit thin space has to be added in lists and intervals: `$(x,\,y)$`, `$[0,\,1]$`. `\StandardMathComma` switches back to the standard behaviour of the comma in French.

   The `icomma` package is an alternative workaround.

9. 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, which should be loaded *after* Babel, see numprint.pdf for more information.

10. babel-french has been designed to take advantage of the xspace package if present: adding \usepackage{xspace} in the preamble will force macros like \fg, \ier, \ieme, \dots, …, to respect the spaces you type after them, for instance typing '1\ier juin' will print '1$^{\text{er}}$ juin' (no need for a forced space after 1\ier).

1.2 Customisation

Customisation of babel-french relies on command \frenchsetup{} (formerly called \frenchbsetup{}, the latter name will be kept for ever to ensure backwards compatibility), options are entered using the l3keys syntax. The command \frenchsetup{} is to appear in the preamble only (after loading Babel).

1.2.1 \frenchsetup{options}

\frenchbsetup{} and \frenchsetup{} are synonymous; the latter should be preferred as the language name for French in Babel is no longer frenchb but french. \frenchsetup{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 l3keys syntax, boolean options (as ShowOptions) can be entered as ShowOptions=true or just ShowOptions, the =true part can be omitted.

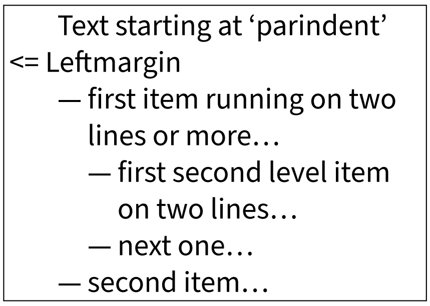
The other options are listed below. Their default value is shown between braces, sometimes followed be a '*'. The '*' means that the default shown applies when babel-french is loaded as the *last* option of Babel —Babel's *main language*—, and is toggled otherwise.

StandardLayout=true (false*) forces babel-french 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; it useless unless French is the main language. This option can be used to avoid conflicts with classes or packages which customise lists or footnotes.

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

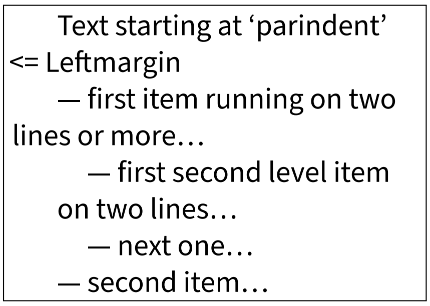
PartNameFull=false (true) ; when true (the default), babel-french numbers the title of \part{} commands as "Première partie", "Deuxième partie" and so on. With some classes which change the \part{} command (AMS classes do so), you could get "Première partie 1", "Deuxième partie 2" in the toc; when this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

`ListItemsAsPar=true (false)` setting this option to `true` is recommended: list items will be displayed as paragraphs with indented labels (in the "Imprimerie Nationale" way) instead of having labels hanging into the left margin. How these two layouts differ is shown below:



```
     Text starting at 'parindent'
<= Leftmargin
     — first item running on two
        lines or more...
        — first second level item
           on two lines...
        — next one...
     — second item...
```

Default French layout



```
     Text starting at 'parindent'
<= Leftmargin
     — first item running on two
lines or more...
         — first second level item
     on two lines...
         — next one...
     — second item...
```

With `ListItemsAsPar=true`

`StandardListSpacing=true (false*)` [6]; `babel-french` usually customises the vertical spaces in the `list` environment, this affects all lists, including `itemize`, `enumerate`, `description`, but also `abstract`, `quote`, `quotation`, `verse`, etc. which are based on `list`. Setting this option to `true` reverts to the standard settings of the `list` environment as defined by the document class.

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

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

`StandardItemLabels=true (false*)` when set to `true` this option prevents `babel-french` 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 `\ding{43}` requires loading the `pifont` package.

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

[6] This option should be used instead of former option `ReduceListSpacing` (kept for backward compatibility) which could be misleading: with some classes (smfart, smfbook f.i.) you had to set `ReduceListSpacing=false` to revert to the class settings which actually reduce list's spacings even more than `babel-french`! `StandardListSpacing=true` replaces `ReduceListSpacing=false`.

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

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

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

`StandardLists=true (false*)` forbids `babel-french` to customise any kind of list. The option `StandardLists=true` should be used in case of conflicts with classes or packages that customise lists too. This option is just a shorthand setting all four options `StandardListSpacing=true`, `StandardItemizeEnv=true`, `StandardEnumerateEnv=true` and `StandardItemLabels=true`.

`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.

`FrenchFootnotes=false (true*)` reverts to the standard layout of footnotes. By default `babel-french` 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 `babel-french` adds a (customisable) 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). The default definition of this thin space is:
`\newcommand*{\FBfnmarkspace}{\kern .5\fontdimen2\font}`

`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 `babel-french` is to automatically typeset non-breaking spaces the width of which is either `\FBthinspace` (defauts to a thin space) before '`;`' '`!`' '`?`' or `\FBcolonspace` (defauts 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, no spurious space is added in that case [7], so the default behaviour of `babel-french` in that area should be fine in most circumstances.

Choosing `AutoSpacePunctuation=false` will ensure that a proper space is added before '`:;!?`' *if and only if* a (normal) space has been typed in. This option gives full control on space insertion before '`:;!?`'. Those who are unsure about their typing in this area should stick to the default option and use

[7] Unless option `OriginalTypewriter` is set, `\ttfamily` is redefined in French to switch off space tuning, see below.

the provided `\NoAutoSpacing` command inside a group in case an unwanted space is added by `babel-french` (i.e. `{\NoAutoSpacing http://mysite}` or `{\NoAutoSpacing ???}` (needed for pdfTeX only).

`ThinColonSpace=true (false)` changes the non-breaking 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'.

`OriginalTypewriter=true (false)` prevents any customisation of `\ttfamily` and `\texttt{}` in French. This option should only be used to ensure backward compatibility. The current default behaviour is to switch off any addition of space before high punctuation with typewriter fonts (*e.g.* verbatim).

`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 `\frquote{}`. This option tells `babel-french` which characters are opening and closing French guillemets (they depend on the input encoding), then you can type either `« guillemets »` or `«guillemets»`[8] (with or without spaces) to get properly typeset French quotes. This option works with XeLaTeX and with pdfLaTeX (default encoding: utf8); with pdflatex other 8-bits encodings (latin1, latin9, ansinew, applemac,...) are also supported when properly declared with `inputenc`.

`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). `babel-french`'s default setting produces slightly narrower spaces with less stretchability.

`EveryParGuill=open, close, none (open)`; sets whether an opening quote («) or a closing one (») or nothing should be printed by `\frquote{}` at the beginning of every parapraph included in a level 1 (outer) quotation. This option is also considered for level 2 (inner) quotations when `InnerGuillSingle=true` (see below).

`InnerGuillSingle=true (false)`; if `InnerGuillSingle=false` (the default), inner quotations entered with `\frquote{}` start with ``` `` ``` and end with `''`. If `InnerGuillSingle=true`, ‹ and › are used instead of British double quotes; moreover if option `EveryParGuill=open` (or `close`) is set, a ‹ (or ›) is added at the beginning of every parapraph included in the inner quotation.

`ThinSpaceInFrenchNumbers=true (false)`; if `numprint` has been loaded with the `autolanguage` option, while typesetting numbers with the `\numprint{}` command, `\npthousandsep` is defined as a non-breaking space (`~`)[9] in French; when set to true, this option redefines `\npthousandsep` as a thin space (`\FBthinspace`).

[8] *Do not* code `«~guillemets~»`.

[9] Actually without stretch nor shrink.

`SmallCapsFigTabCaptions=false (true*)` ; when set to `false`, `\figurename` and `\tablename` will be printed in French captions as “Figure” and “Table” instead of being printed in small caps (the default). The same result can be achieved by defining `\FBfigtabshape` as `\relax` before loading `babel-french` (in a document class f.i.).

`CustomiseFigTabCaptions=false (true*)` ; when set to `false` the default separator (colon) is used instead of `\CaptionSeparator`. Anyway, `babel-french` tries hard to insert a proper space before it in French and warns if it fails to do so.

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

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

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

`SuppressWarning=true (false)` ; can be turned to `true` if you are bored with `babel-french`'s warnings; use this option as *first* option of `\frenchsetup{}` to cancel warnings launched by other options.

Options' order – Please remember that options are read in the order they appear in the `\frenchsetup{}` command. Someone wishing that `babel-french` leaves the layout of lists and footnotes untouched but caring for indentation of first paragraph of sections should choose `\frenchsetup{StandardLayout,IndentFirst}`
The reverse order `\frenchsetup{IndentFirst,StandardLayout}` would lead to option `IndentFirst` being overwritten by `StandardLayout`.

1.2.2 Caption names

All caption names can easily 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 `frenchb` or `francais`.

1.2.3 Figure and table captions

In French, captions in figures and tables should never be printed as ‘Figure 1: ’ which is the default in standard LaTeX2e classes (a space should *always* preceed a

colon in French), anyway 'Figure 1 – ' is preferred.
When French is the main language, the default behaviour of babel-french is to change the separator (colon) used in figures' and tables' captions *for all languages* to \CaptionSeparator which defaults to ' – ' and can be redefined in the preamble with \renewcommand*{\CaptionSeparator}{...}. This works for the standard LaTeX2e classes, for the memoir koma-script and beamer classes. In case this procedure fails a warning is issued.
When French is not the main language, the colon is preserved for all languages including French but babel-french tries hard to insert a proper space before it and warns if it fails to do so.
Three options are provided to customise figure and table captions:

- CustomiseFigTabCaptions is set to true when French is the main language (hence separator = ' – ') and to false otherwise (hence separator = ': ' with a proper space before the colon in French if possible); toogle this option if needed;

- the second option, OldFigTabCaptions, can be set to true to print figures' and tables' captions as they were with versions pre 3.0 of babel-french (using \CaptionSeparator in French and colon in other languages); this option only makes sense with the standard LaTeX classes article, report and book;

- the last option, SmallCapsFigTabCaptions, can be set to false to typeset \figurename and \tablename in French as "Figure" and "Table" rather than in small caps (the default).

1.3 Hyphenation checks

Once you have built your format, a good precaution would be to perform some basic tests about hyphenation in French. For LaTeX2e I suggest this:

- run pdfLaTeX on the following file:

```
%%% Test file for French hyphenation.
\documentclass[french]{article}
\usepackage[utf8]{inputenc} % utf8, what else?
\usepackage[T1]{fontenc}    % mandatory for French
\usepackage{lmodern}        % or erewhon, palatino...
\usepackage{babel}
\begin{document}
\showhyphens{signal container \'ev\'enement alg\`ebre}
\showhyphens{signal container événement algèbre}
\end{document}
```

- check the hyphenations proposed by TeX in your log-file; in French you should get with both 7-bit and 8-bit encodings
si-gnal contai-ner évé-ne-ment al-gèbre.

> Do not care about how accented characters are displayed in the log-file, what matters is the position of the ‘`-`’ hyphen signs *only*.

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

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

1.4 Changes

What's new in version 3.7?

The acadian dialect is no longer supported: coding \usepackage[acadian]{babel} prints a warning and uses french instead. Reason: I have never got feedback from anybody using them; anyway babel-french is customisable enough to fit any French dialect, see \fbsetup{} p.44.
babel-french has been split into two files frenchb3.dtx (this file) the legacy part which is frozen, meant for TeX, pdfTeX and XeTeX engines *only*, and frenchb.dtx meant for LuaTeX.
Option GlobalLayoutFrench has been deleted: it doesn't make sense to change the lists' layout at language switches.
Note on PDF tagging: this project requires a complete redesign of lists based on templates. The new lists templates, still experimental, are incompatible with babel-french lists' customisation, which is consequently disabled when tagging is enabled. A warning is issued in the .log. See https://github.com/latex3/tagging-project/issues/694 for more information. I plan to get babel-french lists' customisation working again asap (hopefully with the next LaTeX release 2025/10/01).
Version 3.7b takes advantage of the new footnotes' template (when it is available) to customise the footnotes' layout. This should fix issue 932.

What's new in version 3.6?

Version 3.6a no longer loads the keyval package, replaced by core LaTeX commands (l3keys). The thin space added before footnote's calls is now customisable (suggested by Thomas Savary), the command's name is \FBfnmarkspace.

What's new in version 3.5?

Version 3.5a offers a new option ListItemsAsPar. The default layout of lists is unchanged (for backward compatibility), but users should try this new option which ensures a layout of lists closer to French typographic standards: see f.i. how lists are typeset in the book "Lexique des règles typographiques en usage à l'Imprimerie Nationale".
Version 3.5b fixes a bug due to wrong \everypar's management in \frquote{}; it showed up when \frquote{} immediately followed a sectionning command.
Starting with version 3.5d, a new option StandardListSpacing has been added to supersede ReduceListSpacing.
A new command \NoEveryParQuote has been added in version 3.5e: it is meant to be used inside a group or environment to suppress unwanted guillemets (typically when lists are embedded in \frquote{}).
Version 3.5g fixes a long standing bug affecting LuaTeX: legacy kerning was disabled for Type1 fonts since v3.1g (2015).

Version 3.5j also fixes a long standing bug affecting koma-script, `memoir` and `beamer` classes: redefintions of the caption separator (commands `\captionformat`, `\captiondelim`, etc.) are now taken into account properly.
Version 3.5k is a cleanup release:

- the translations in French of `\figurename` and `\tablename` no longer hold font changing commands (switch to small caps), the font switch has been moved to `\fnum@figure` and `\fnum@table` as suggested by Axel Sommerfeldt.
- Package `caption` can now be loaded whether before or after `babel`, indifferently.
- `\pdfstringdefDisableCommands` is no longer used: as suggested by the LaTeX3 team, all commands requiring special care in `hyperref`'s bookmarks are now defined using `\textorpdfstring{}{}`.

Version 3.5n introduces a new command `\bname{}` (an alternative to `\bsc{}`).
Version 3.5q corrects a bug in lists layout: `\listparindent` (formely 0pt) is defined as `\parindent` and if `\parskip > 0pt`, `\parsep` is now defined as `\parskip`. This ensures that paragraphs included in lists are now visible. The former behaviour can be recovered by adding `\parskip=0pt`, `\parindent=0pt` *inside* the list environment.
Version 3.5r is compatible with `ucharclasses` which is now loaded by `fontsetup` with the XeTeX engine. The `frenchb.ins` file is no longer needed to extract the `.ldf` files from `frenchb.dtx` (see `README.md`).

What's new in version 3.4?

Version 3.4a adds a new command `\frenchdate` (see p. 4) and slightly changes number formatting: `\FBthousandsep` is now a *kern* instead of a rubber length. `\renewcommand*{\FBthousandsep}{~}` will switch back to the former (wrong) behaviour.
A new command `\FBsetspaces` has been added for easy customising of spacing before high punctuation and inside quotes, see p. 19.
Version 3.4 requires eTeX and LuaTeX 1.0.4 or newer.

What's new in version 3.3?

In version 3.3d the automatic insertion of non-breaking spaces before the colon character has been improved *with engine LuaTeX only*: a spurious space is no longer inserted in strings like http://mysite, C:\Program Files or 10:55. Unfortunately, my attempts to do the same with XeTeX or pdfTeX were unsuccessful.
A few internal changes have been made in version 3.3c to improve the convertion into HTML of non-breaking spaces added by `babel-french`. Usage of `lwarp` (v.0.37 and up) is recommended for HTML output, it works fine on files compiled with XeLaTeX or pdfLaTeX formats.

According to current `Babel`'s standards, every dialect should have it's own `.ldf` file; starting with version 3.3b, the main support for French is in `french.ldf`, portmanteau files `frenchb.ldf`,`francais.ldf`, `acadian.ldf` and `canadien.ldf` have been added. The only recommended option is `french` all other are deprecated.
Release 3.3a is compatible with LuaTeX v. 0.95 (TL2016) and up. Former skips `\FBcolonskip`, `\FBthinskip` and `\FBguillskip` controlling punctuation spacings in LuaTeX have been removed; all three engines now rely on the same commands `\FBcolonspace`, `\FBthinspace` and `\FBguillspace`.
Further customisation of the `\part{}` command is provided via three new commands `\frenchpartfirst`, `\frenchpartsecond` and `\frenchpartnameord`.

2 The code

2.1 Initial setup

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

```
<*french>
\LdfInit\CurrentOption{FBclean@on@exit}
```

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

```
\def\fb@error#1#2{%
    \begingroup
      \newlinechar=`\^^J
      \def\\{^^J(french3.ldf) }%
      \errhelp{#2}\errmessage{\\#1^^J}%
    \endgroup}
\def\fb@warning#1{%
   \begingroup
      \newlinechar=`\^^J
      \def\\{^^J(french3.ldf) }%
      \message{\\#1^^J}%
   \endgroup}
\def\fb@info#1{%
   \begingroup
      \newlinechar=`\^^J
      \def\\{^^J}%
      \wlog{#1}%
   \endgroup}
```

Quit if eTeX is not available.

```
\let\bbl@tempa\relax
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname eTeXversion\endcsname\relax
  \let\bbl@tempa\endinput
  \fb@error{babel-french requires eTeX.\\
            Aborting here}
           {Orignal PlainTeX is not supported,\\
            please use LuaTeX or XeTeX engines.}
\fi
\bbl@tempa
```

Quit if `Babel`’s version is less than 3.9i.

```
\let\bbl@tempa\relax
\ifdefined\babeltags
\else
   \let\bbl@tempa\endinput
   \ifdefined\PackageError
      \PackageError{french.ldf}
         {babel-french requires babel v.3.16.\MessageBreak
          Aborting here}
         {Please upgrade Babel!}
   \else
      \fb@error{babel-french requires babel v.3.16.\\
                Aborting here}
               {Please upgrade Babel!}
   \fi
\fi
\bbl@tempa
```

Make sure that `\l@french` is defined (fallbacks are `\l@nohyphenation` if available or 0). `babel.def` (3.9i and up) defines `\l@<languagename>` also for eTeX and XeTeX formats which set `\lang@<languagename>`.

```
\def\FB@nopatterns{%
   \ifdefined\l@nohyphenation
      \adddialect\l@french\l@nohyphenation
      \edef\bbl@nulllanguage{\string\language=nohyphenation}%
   \else
      \edef\bbl@nulllanguage{\string\language=0}%
      \adddialect\l@french0
   \fi
   \@nopatterns{French}}
\ifdefined\l@french \else \FB@nopatterns \fi
```

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

```
\providehyphenmins{french}{\tw@\thr@@}
```

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

```
\newif\ifLaTeXe
\let\bbl@tempa\relax
\ifdefined\magnification
\else
   \ifdefined\@compatibilitytrue
     \LaTeXetrue
   \else
     \PackageError{french3.ldf}
        {LaTeX-2.09 format is no longer supported.\MessageBreak
         Aborting here}
        {Please upgrade to LaTeX2e!}
     \let\bbl@tempa\endinput
   \fi
\fi
```

```
\bbl@tempa
```

\ifFBXeTeX \ifFBunicode French hyphenation patterns are now coded in Unicode, see file `hyph-fr.tex`. XeTeX and LuaTeX engines require some extra code to deal with the French “apostrophe”. Let’s define a new ‘if’: `\ifFBunicode` which will be true for the XeTeX engine and false for 8-bits engines. `\ifFBXeTeX` is kept for backward compatibility only.

```
\newif\ifFBXeTeX
\newif\ifFBunicode
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname XeTeXrevision\endcsname\relax
\else
  \FBunicodetrue \FBXeTeXtrue
\fi
```

\ifFBfrench True when the current language is French or any of its dialects; will be set to true by `\extrasfrench` and to false by `\noextrasfrench`. Used in `\DecimalMathComma` and `frenchsetup{og=«, fg=»}`.

```
\newif\ifFBfrench
```

\extrasfrench \noextrasfrench 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” (U+27 or U+2019) 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 XeTeX, and pdfTeX using `hyph-fr.tex` patterns.

```
\def\extrasfrench{%
     \FBfrenchtrue
     \babel@savevariable{\lccode"27}%
     \lccode"27="27
     \ifFBunicode
       \babel@savevariable{\lccode"2019}%
       \lccode"2019="2019
     \fi
}
\def\noextrasfrench{\FBfrenchfalse}
```

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

```
\addto\extrasfrench{\bbl@frenchspacing}
\addto\noextrasfrench{\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. XeTeX provides 'XeTeXinterchar' as an alternative to active characters.

\ifFB@active@punct

Three internal flags are needed for the three different techniques used for 'high punctuation' management.

```
\newif\ifFB@active@punct \FB@active@puncttrue
```

\ifFB@xetex@punct

For XeTeX, the availability of \XeTeXinterchartokenstate decides whether the 'high punctuation' characters (; ! ? and :) have to be made \active or not.
The number of available character classes has been increased from 256 to 4096 in XeTeX v. 0.99994, the class for non-characters is now 0xFFF=4095 (formerly 0xFF=255). The class for standard characters is 0.

```
\newcount\FB@stdchar
\newif\ifFB@xetex@punct
\ifdefined\XeTeXinterchartokenstate
  \FB@xetex@puncttrue\FB@active@punctfalse
  \ifdim\the\XeTeXversion\XeTeXrevision\p@ < 0.99994\p@
    \chardef\FB@nonchar="FF \relax
  \else
    \chardef\FB@nonchar="FFF \relax
  \fi
  \FB@stdchar=\z@
\fi
```

\FBguillspace
\FBcolonspace
\FBthinspace

These three commands are meant for basic French. Other French dialects can use different settings, see below. According to the I.N. specifications, the ':' requires an inter-word space before it, the other three require just a thin space. We define \FBcolonspace as \space (inter-word space) and \FBthinspace as an half inter-word space with no shrink nor stretch. \FBguillspace is defined btw. as spacing for French quotes. \FBguillspace has been fine tuned by Thierry Bouche to 80% of an inter-word space with reduced stretchability. All three are user customisable in the preamble, best using the \FBsetspaces command described below. A penalty will be added before these spaces to prevent line breaking.

```
\newcommand*{\FBguillspace}{\hskip .8\fontdimen2\font
                             plus .3\fontdimen3\font
                            minus .8\fontdimen4\font \relax}
\newcommand*{\FBcolonspace}{\space}
\newcommand*{\FBthinspace}{\hskip .5\fontdimen2\font \relax}
```

\FBsetspaces

This command makes it easy to fine tune \FBguillspace, \FBcolonspace and \FBthinspace in French (defaut) or independently in a French dialect using the optional argument. They are meant for LaTeX2e *only* and can only be used in the preamble. Four mandatory arguments[10]: the first one is a *string* either ”guill”, ”colon”, or ”thin”, the last three are decimal numbers specifying *width*, *stretch* and *shrink* relative to the relevant *fontdimens*. For instance `\FBsetspaces{colon}{0.5}{0}{0}` defines `\FBcolonspace` as a thinspace as suggested by the “Guide du typographe Roman’’.

```
\ifLaTeXe
  \newcommand*{\FBsetspaces}[5][french]{%
    \@namedef{FB#2space}{\hskip #3\fontdimen2\font
                             plus #4\fontdimen3\font
                            minus #5\fontdimen4\font \relax}}%
  \@onlypreamble\FBsetspaces
\fi
```

The conditional `\ifFB@spacing` will be used by pdfTeX and XeTeX engines to switch on or off space tuning before high punctuation and inside French quotes.

```
\newif\ifFB@spacing \FB@spacingtrue
```

2.2.1 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 `\frenchsetup{}` (see section 2.11).
Unless `ucharclass` is loaded, the default value for `\XeTeXcharclass` is 0 for characters tokens and `\FB@nonchar` for all other tokens (glues, kerns, math and box boundaries, etc.). `ucharclass` defines a XeTeX class for every range of Unicode characters in order to facilitate font switching. Most French characters belong to range [”20, ”7F] (class `\BasicLatinClass`) some (accented chars, diacritics,…) to range [”80, ”FF] (class `\LatinSupplementClass`) and three (œ, Œ, and the long-s) to [”100, ”17F] (class `\LatinExtendedAClass`).
We check AtBeginDocument whether `ucharclass` is loaded; if so, when switching to French, the class `\FB@stdchar` of all characters possibly used in French (except punctuation) will be forced to `\BasicLatinClass` which is the default for most of them, the class of the others (accented chars, ligatures, diacritics, etc.) will be saved and changed locally in French, then restored to their original value when leaving French.
We switch `\XeTeXinterchartokenstate` to 1 and change the `\XeTeXcharclass` values of `;` `!` `?` `:` `(` `]` « and » when entering French. Their inital values will be restored when leaving French.
The following part holds specific code for punctuation with XeTeX engines.

```
\newif\ifFB@og@fg@xetex
```

[10] The former optional `lang` argument no longer has any effect.

```
\ifFB@xetex@punct
  \ifLaTeXe
    \PackageInfo{french3.ldf}{No need for active punctuation
                             characters\MessageBreak with this
                             version of XeTeX!\MessageBreak reported}
  \else
    \fb@info{No need for active punctuation characters\\
             with this version of XeTeX!}
  \fi
```

Six new character classes are defined for babel-french.

```
  \newXeTeXintercharclass\FB@punctthick
  \newXeTeXintercharclass\FB@punctthin
  \newXeTeXintercharclass\FB@punctnul
  \newXeTeXintercharclass\FB@guilo
  \newXeTeXintercharclass\FB@guilf
  \newXeTeXintercharclass\FB@guilnul
```

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

```
  \def\FBsavevariable@loop#1#2{\begingroup
    \toks@\expandafter{\originalTeX #1}%
    \edef\x{\endgroup
      \def\noexpand\originalTeX{\the\toks@ #2=\the#1#2\relax}}%
    \x}
```

\FB@charlistsave holds the all list of characters which have their \XeTeXcharclass value modified in French: it always includes high punctuation, French quotes, opening delimiters and no-break spaces. If ucharclasses is loaded, non-ASCII characters used in French have to be added; as xeCJK changes the class of some characters used in French, these have to be saved too if xeCJK is loaded.

```
  \def\FB@charlist{"21,"3A,"3B,"3F,"AB,"BB,"28,"5B,"A0,"202F}
  \def\FB@charlistUCC{}
  \def\FB@charlistxeCJK{}
  \edef\FB@charlistsave{\FB@charlist}
  \ifLaTeXe
     \AddToHook{env/document/before}{%
       \IfPackageLoadedTF{ucharclasses}%
         {\ifdefined\BasicLatinClass
            \RenewCommandCopy{\FB@stdchar}{\BasicLatinClass}%
            \def\FB@charlistUCC{"C0,"C2,"C6,"C7,"C8,"C9,"CA,"CB,"CE,"CF,%
              "D4,"D6,"D9,"DB,"DC,"E0,"E2,"E6,"E7,"E8,"E9,"EA,"EB,"EE,%
              "EF,"F4,"F6,"F9,"FB,"FC,"152,"153,"17F,"2019}%
            \addto\FB@charlist{,\FB@charlistUCC}%
            \edef\FB@charlistsave{\FB@charlist}%
          \fi
         }{}%
       \IfPackageLoadedTF{xeCJK}%
         {\def\FB@charlistxeCJK{%
                  "29,"5D,"7B,"7D,"2C,"2D,"2E,"22,"25,"27,"60,"2019}%
          \addto\FB@charlist{,\FB@charlistxeCJK}%
          \edef\FB@charlistsave{\FB@charlist}%
         }{}%
     }
  \fi
```

\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.

```
\newcommand*{\FB@xetex@punct@french}{%
  \babel@savevariable{\XeTeXinterchartokenstate}%
  \bbl@for\FB@char\FB@charlistsave
       {\FBsavevariable@loop{\XeTeXcharclass}{\FB@char}}%
```

If `ucharclasses` is loaded, force non-ASCII used in French to class `\FB@stdchar` (=`\BasicLatinClass`).

```
  \ifx\FB@charlistUCC\@empty\else
    \bbl@for\FB@char\FB@charlistUCC
      {\XeTeXcharclass \FB@char \FB@stdchar}%
  \fi
```

These characters have their class changed by `xeCJK.sty`, let’s reset their class in French.

```
  \ifx\FB@charlistxeCJK\@empty\else
    \bbl@for\FB@char\FB@charlistxeCJK
            {\XeTeXcharclass\FB@char=\FB@stdchar}%
  \fi
```

Assign classes related to French double quotes if options `og=«, fg=»` have been selected.

```
  \ifFB@og@fg@xetex
    \XeTeXcharclass"13   = \FB@guilo
    \XeTeXcharclass"14   = \FB@guilf
    \XeTeXcharclass"AB   = \FB@guilo
    \XeTeXcharclass"BB   = \FB@guilf
    \XeTeXcharclass"A0   = \FB@guilnul
    \XeTeXcharclass"202F = \FB@guilnul
  \fi
```

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

```
  \bbl@for\FB@char {`\[,`\(,"A0,"202F}%
          {\XeTeXcharclass\FB@char=\FB@punctnul}%
```

Let’s now define specific classes for punctuation and interactions between classes. When false, the flag `\ifFB@spacing` switches off any interaction between classes (this flag is controlled by user-level command `\NoAutoSpacing`; this flag is also set to false when the current font is a typewriter font).

```
\XeTeXinterchartokenstate=\@ne
\XeTeXcharclass `\: = \FB@punctthick
\XeTeXinterchartoks \FB@stdchar \FB@punctthick = {%
       \ifFB@spacing\ifhmode\FDP@colonspace\fi\fi}%
\XeTeXinterchartoks \FB@guilf \FB@punctthick = {%
       \ifFB@spacing\FDP@colonspace\fi}%
```

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

```
\XeTeXinterchartoks \FB@nonchar \FB@punctthick = {%
   \ifFB@spacing
     \ifhmode
       \ifdim\lastskip>1sp
         \unskip\penalty\@M\FBcolonspace
       \else
         \FDP@colonspace
       \fi
     \fi
   \fi}%
\bbl@for\FB@char {`\;,`\!,`\?}%
        {\XeTeXcharclass\FB@char=\FB@punctthin}%
\XeTeXinterchartoks \FB@stdchar \FB@punctthin = {%
       \ifFB@spacing\ifhmode\FDP@thinspace\fi\fi}%
\XeTeXinterchartoks \FB@guilf \FB@punctthin = {%
       \ifFB@spacing\FDP@thinspace\fi}%
\XeTeXinterchartoks \FB@nonchar \FB@punctthin = {%
       \ifFB@spacing
         \ifhmode
           \ifdim\lastskip>1sp
             \unskip\penalty\@M\FBthinspace
           \else
             \FDP@thinspace
           \fi
         \fi
       \fi}%
\ifFB@og@fg@xetex
  \XeTeXinterchartoks \FB@guilo \FB@stdchar = {%
       \ifFB@spacing\FB@guillspace\fi}%
  \XeTeXinterchartoks \FB@guilo \FB@nonchar = {%
       \ifFB@spacing\FB@guillspace\ignorespaces\fi}%
  \XeTeXinterchartoks \FB@stdchar \FB@guilf = {%
```

```
           \ifFB@spacing\FB@guillspace\fi}%
      \XeTeXinterchartoks \FB@puncthin \FB@guilf = {%
           \ifFB@spacing\FB@guillspace\fi}%
      \XeTeXinterchartoks \FB@nonchar \FB@guilf = {%
           \ifFB@spacing\unskip\FB@guillspace\fi}%
    \fi
   }
   \addto\extrasfrench{\FB@xetex@punct@french}
```

End of specific code for punctuation with modern XeTeX engines.

```
\fi
```

2.2.2 Punctuation with standard (pdf)TeX

In standard (pdf)TeX we need to make the four characters `;` `!` `?` and `:` ‘active’ and provide their definitions. Before doing so, we have to save some definitions involving `:`.

```
\newif\ifFB@koma
\ifLaTeXe
  \@ifclassloaded{scrartcl}{\FB@komatrue}{}
  \@ifclassloaded{scrbook}{\FB@komatrue}{}
  \@ifclassloaded{scrreprt}{\FB@komatrue}{}
  \ifFB@koma\def\FB@std@capsep{:\ }\fi
  \@ifclassloaded{beamer}{\def\FB@std@capsep{:\ }}{}
  \@ifclassloaded{memoir}{\def\FB@std@capsep{: }}{}
\fi
```

```
\ifFB@active@punct
  \initiate@active@char{:}%
  \initiate@active@char{;}%
  \initiate@active@char{!}%
  \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 a non-breaking `\FBthinspace` instead. If no space has been typed, we add `\FDP@thinspace` which will be defined, up to the user’s wishes, as a non-breaking `\FBthinspace` or as `\@empty`.

```
  \declare@shorthand{french}{;}{%
    \ifFB@spacing
      \ifhmode
        \ifdim\lastskip>1sp
          \unskip\penalty\@M\FBthinspace
        \else
          \FDP@thinspace
      \fi
    \fi
  \fi
```

Now we can insert a ; character.

```
  \string;}
```

The next three definitions are very similar.

```
\declare@shorthand{french}{!}{%
  \ifFB@spacing
    \ifhmode
      \ifdim\lastskip>1sp
        \unskip\penalty\@M\FBthinspace
      \else
        \FDP@thinspace
      \fi
    \fi
  \fi
  \string!}
\declare@shorthand{french}{?}{%
  \ifFB@spacing
    \ifhmode
      \ifdim\lastskip>1sp
        \unskip\penalty\@M\FBthinspace
      \else
        \FDP@thinspace
      \fi
    \fi
  \fi
  \string?}
\declare@shorthand{french}{:}{%
  \ifFB@spacing
    \ifhmode
      \ifdim\lastskip>1sp
        \unskip\penalty\@M\FBcolonspace
      \else
        \FDP@colonspace
      \fi
    \fi
  \fi
  \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.

```
\declare@shorthand{system}{:}{\string:}
\declare@shorthand{system}{!}{\string!}
\declare@shorthand{system}{?}{\string?}
```

```
  \declare@shorthand{system}{;}{\string;}
```

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

```
  \addto\extrasfrench{\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.

```
    \bbl@activate{:}\bbl@activate{;}%
    \bbl@activate{!}\bbl@activate{?}%
  }
  \addto\noextrasfrench{%
    \bbl@deactivate{:}\bbl@deactivate{;}%
    \bbl@deactivate{!}\bbl@deactivate{?}%
  }
\fi
```

2.2.3 Punctuation switches common to both engines

A new ‘if’ \ifFBAutoSpacePunctuation needs to be defined now to control the two possible ways of dealing with ‘high punctuation’. it’s default value is true, but it can be set to false by \frenchsetup{AutoSpacePunctuation=false} for finer control.

```
\newif\ifFBAutoSpacePunctuation  \FBAutoSpacePunctuationtrue
```

\AutoSpaceBeforeFDP
\NoAutoSpaceBeforeFDP

\autospace@beforeFDP and \noautospace@beforeFDP are internal commands. \autospace@beforeFDP defines commands \FDP@thinspace and \FDP@colonspace as non-breaking spaces, while \noautospace@beforeFDP makes them no-op. User commands \AutoSpaceBeforeFDP and \NoAutoSpaceBeforeFDP do the same and take care of the flag \ifFBAutoSpacePunctuation in LaTeX.
Set the default now for Plain (done later for LaTeX).

```
\def\autospace@beforeFDP{%
  \def\FDP@thinspace{\penalty\@M\FBthinspace}%
  \def\FDP@colonspace{\penalty\@M\FBcolonspace}}
\def\noautospace@beforeFDP{%
  \let\FDP@thinspace\@empty
  \let\FDP@colonspace\@empty}
\ifLaTeXe
  \def\AutoSpaceBeforeFDP{\autospace@beforeFDP
                          \FBAutoSpacePunctuationtrue}
  \def\NoAutoSpaceBeforeFDP{\noautospace@beforeFDP
                            \FBAutoSpacePunctuationfalse}
  \AtEndOfPackage{\AutoSpaceBeforeFDP}
\else
  \let\AutoSpaceBeforeFDP\autospace@beforeFDP
  \let\NoAutoSpaceBeforeFDP\noautospace@beforeFDP
  \AutoSpaceBeforeFDP
\fi
```

\rmfamilyFB \sffamilyFB \ttfamilyFB In LaTeX2e \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. When AutoSpacePunctuation is false, the eventually typed spaces are left unchanged (not turned into thin spaces, no penalty added). \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 \frenchsetup{}. This is also a workaround for the weird behaviour of these characters in verbatim mode.

```
\ifLaTeXe
  \DeclareRobustCommand\ttfamilyFB{\FB@spacingfalse \ttfamilyORI}
  \DeclareRobustCommand\rmfamilyFB{\FB@spacingtrue  \rmfamilyORI}
  \DeclareRobustCommand\sffamilyFB{\FB@spacingtrue  \sffamilyORI}
\fi
```

\NoAutoSpacing The following command disables automatic spacing for high punctuation and French quote characters; it also switches off active punctuation characters (if any). It is engine independent (works for TeX, pdfTeX and XeTeX based engines) and is meant to be used inside a group. The faked definition of \texorpdfstring will be overwritten by hyperref.sty.

```
\providecommand\texorpdfstring[2]{#1}
\DeclareRobustCommand*{\NoAutoSpacing}{%
  \texorpdfstring{\FB@spacingfalse
                  \ifFB@active@punct\shorthandoff{;:!?}\fi}{}%
}
```

2.3 Commands for French quotation marks

\guillemotleft \guillemotright \textquoteddblleft \textquoteddblright pdfLaTeX users are supposed to use 8-bit output encodings (T1, LY1,...) to typeset French, those who still stick to OT1 should load 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 \guillemotleft and \guillemotright are defined by package fontspec (v. 2.5d and up).
We provide the following definitions for non-LaTeX users only as fall-back, they are welcome to change them for anything better.

```
\ifLaTeXe
\else
  \ifFBunicode
    \def\guillemotleft{{\char"00AB}}
    \def\guillemotright{{\char"00BB}}
    \def\textquotedblleft{{\char"201C}}
    \def\textquotedblright{{\char"201D}}
  \else
    \def\guillemotleft{\leavevmode\raise0.25ex
                       \hbox{$\scriptscriptstyle\ll$}}
    \def\guillemotright{\raise0.25ex
                        \hbox{$\scriptscriptstyle\gg$}}
    \def\textquotedblleft{``}
    \def\textquotedblright{''}
  \fi
  \let\xspace\relax
\fi
```

\FB@og \FB@fg The next step is to provide correct spacing after '«' and before '»'; no line break is allowed neither *after* the opening one, nor *before* the closing one. French quotes (including spacing) are printed by \FB@og and \FB@fg, the expansion of the top level commands \og and \fg is different in and outside French.
\FB@og and \FB@fg are now designed to work in bookmarks.

```
\newcommand*{\FB@og}{\texorpdfstring{\@FB@og}{\guillemotleft\space}}
\newcommand*{\FB@fg}{\texorpdfstring{\@FB@fg}{\space\guillemotright}}
```

The internal definitions \@FB@og and \@FB@fg need some engine-dependent tuning. With XeTeX, \ifFB@spacing is set to false locally to prevent the quotes characters from adding space when option og=«, fg=» is set.

```
\newcommand*{\FB@guillspace}{\penalty\@M\FBguillspace}
\ifFB@xetex@punct
  \DeclareRobustCommand*{\@FB@og}{\leavevmode
        \bgroup\FB@spacingfalse\guillemotleft\egroup
        \FB@guillspace}
  \DeclareRobustCommand*{\@FB@fg}{\ifdim\lastskip>\z@\unskip\fi
        \FB@guillspace
        \bgroup\FB@spacingfalse\guillemotright\egroup}
\fi
\ifFB@active@punct
  \DeclareRobustCommand*{\@FB@og}{\leavevmode
        \guillemotleft
        \FB@guillspace}
  \DeclareRobustCommand*{\@FB@fg}{\ifdim\lastskip>\z@\unskip\fi
        \FB@guillspace
        \guillemotright}
\fi
```

\og \fg The user level macros for quotation marks are named \og (“ouvrez guillemets”) and \fg (“fermez guillemets”). Another option for typesetting quotes in French is to use the command \frquote (see below). Dummy definition of \og and \fg just to ensure that this commands are not yet defined.

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

The definitions of \og and \fg for 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. 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).

```
\ifLaTeXe
  \def\bbl@frenchguillemets{%
        \renewcommand*{\og}{\FB@og}%
        \renewcommand*{\fg}{\FB@fg\xspace}}
  \renewcommand*{\og}{\textquotedblleft}
  \renewcommand*{\fg}{\ifdim\lastskip>\z@\unskip\fi
                      \textquotedblright\xspace}
\else
  \def\bbl@frenchguillemets{\let\og\FB@og
                            \let\fg\FB@fg}
  \def\og{\textquotedblleft}
  \def\fg{\ifdim\lastskip>\z@\unskip\fi\textquotedblright}
\fi
```

```
\addto\extrasfrench{\babel@save\og \babel@save\fg
                    \bbl@frenchguillemets}
```

\frquote Another way of entering French quotes relies on \frquote{} with supports up to two levels of quotes. Let’s define the default quote characters to be used for level one or two of quotes…

```
\newcommand*{\ogi}{\FB@og}
\newcommand*{\fgi}{\FB@fg}
\newcommand*{\@ogi}{\ifmmode\hbox{\ogi}\else\ogi\fi}
\newcommand*{\@fgi}{\ifmmode\hbox{\fgi}\else\fgi\fi}
\newcommand*{\ogii}{\ifFBInnerGuillSingle
                      \guilsinglleft\FB@guillspace
                    \else \textquotedblleft
                    \fi}
\newcommand*{\fgii}{\ifFBInnerGuillSingle
                      \ifdim\lastskip>\z@\unskip\fi
                      \FB@guillspace\guilsinglright
                    \else \textquotedblright
                    \fi}
\newcommand*{\@ogii}{\ifmmode\hbox{\ogii}\else\ogii\fi}
\newcommand*{\@fgii}{\ifmmode\hbox{\fgii}\else\fgii\fi}
```

and the needed technical stuff to handle options:

```
\newcount\FBguill@level
\newtoks\FBold@everypar
```

`\FB@addquote@everypar` was borrowed from `csquotes.sty`.

```
\def\FB@addquote@everypar{%
  \let\FBnew@everypar\everypar
  \FBold@everypar=\expandafter{\the\everypar}%
  \FBnew@everypar={\the\FBold@everypar\FBeverypar@quote}%
  \let\everypar\FBold@everypar
  \let\FB@addquote@everypar\relax
}
\newif\ifFBcloseguill  \FBcloseguilltrue
\newif\ifFBInnerGuillSingle
\def\FBguillopen{\bgroup\NoAutoSpacing\guillemotleft\egroup}
\def\FBguillclose{\bgroup\NoAutoSpacing\guillemotright\egroup}
\let\FBguillnone\empty
\let\FBeveryparguill\FBguillopen
\let\FBeverylineguill\FBguillnone
\let\FBeverypar@quote\relax
\let\FBeveryline@quote\empty
```

The main command `\frquote` accepts (in LaTeX2e only) a starred version which suppresses the closing quote; it is meant to be used for inner quotations which end together with the outer one, then only one closing guillemet (the outer one) should be printed. `\frquote` (without star) is now designed to work in bookmarks too.

```
\ifLaTeXe
  \DeclareRobustCommand\frquote{%
    \texorpdfstring{\@ifstar{\FBcloseguillfalse\fr@quote}%
                            {\FBcloseguilltrue \fr@quote}}%
                   {\bm@fr@quote}%
  }
  \newcommand{\bm@fr@quote}[1]{%
    \guillemotleft\space #1\space\guillemotright}
\else
  \newcommand\frquote[1]{\fr@quote{#1}}
\fi
```

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

```
\newcommand{\fr@quote}[1]{%
  \leavevmode
  \advance\FBguill@level by \@ne
  \ifcase\FBguill@level
  \or
```

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

```
    \ifx\FBeveryparguill\FBguillnone
    \else
      \def\FBeverypar@quote{\FBeveryparguill\FB@guillspace}%
      \FB@addquote@everypar
    \fi
    \@ogi #1\@fgi
  \or
```

This for level 2 (inner) quotations. We eventually need to redefine `\FBeverypar@quote` for level 2 quotations:

```
    \let\FBeverypar@quote\relax
    \ifFBInnerGuillSingle
      \ifx\FBeveryparguill\FBguillopen
        \def\FBeverypar@quote{\guilsinglleft\FB@guillspace}%
      \fi
    \fi
    \ifx\FBeveryparguill\FBguillclose
          \def\FBeverypar@quote{\guilsinglright\FB@guillspace}%
    \fi
    \@ogii #1\@fgii
  \else
```

Warn if `\FBguill@level` > 2:

```
    \ifx\PackageWarning\@undefined
      \fb@warning{\noexpand\frquote\space handles up to
                  two levels.\\ Quotation not printed.}%
    \else
      \PackageWarning{french3.ldf}{%
         \protect\frquote\space handles up to two levels.
         \MessageBreak Quotation not printed.  Reported}
    \fi
  \fi
```

Closing: step down `\FBguill@level` and clean on exit. Changes made global in case `\frquote{}` ends inside an environment.

```
  \global\advance\FBguill@level by \m@ne
  \ifcase\FBguill@level \global\let\FBeverypar@quote\relax
  \or \gdef\FBeverypar@quote{\FBeveryparguill\FB@guillspace}%
      \global\let\FBeveryline@quote\empty
  \fi
}
```

The next command is intended to be used in list environments to suppress quotes which might be added by `\FBeverypar@quote` after items for instance.

```
\newcommand*{\NoEveryParQuote}{\let\FBeveryparguill\FBguillnone}
```

2.4 Date in French

\frenchtoday
\frenchdate
\datefrench

The following code creates a macro \datefrench which in turn defines command \frenchtoday (\today is defined as \frenchtoday in French). This new implementation relies on commands \SetString and \SetStringLoop, therefore requires Babel 3.10 or newer.

```
\StartBabelCommands*{french}{date}
    [unicode, fontenc=TU EU1 EU2, charset=utf8]
  \SetString\monthiiname{février}
  \SetString\monthviiiname{août}
  \SetString\monthxiiname{décembre}
\StartBabelCommands*{french}{date}
  \SetStringLoop{month#1name}{%
      janvier,f\'evrier,mars,avril,mai,juin,juillet,%
      ao\^ut,septembre,octobre,novembre,d\'ecembre}
  \SetString\today{\FB@date{\year}{\month}{\day}}
\EndBabelCommands
```

\frenchdate (which produces an unbreakable string) and \frenchtoday (breakable) both rely on \FB@date, the inner group is needed for \hbox.

```
\newcommand*{\FB@date}[3]{%
  {{\number#3}\ifnum1=#3{\ier}\fi\FBdatespace
  \csname month\romannumeral#2name\endcsname
  \ifx#1\@empty\else\FBdatespace\number#1\fi}}
\newcommand*{\FBdatebox}{\hbox}
\newcommand*{\FBdatespace}{\space}
\newcommand*{\frenchdate}{\FBdatebox\FB@date}
```

2.5 Extra utilities

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

\up
\fup

\up eases the typesetting of superscripts like ‘1$^{\text{er}}$’. Up to version 2.0 of babel-french \up was just a shortcut for \textsuperscript in LaTeX2e, 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 \frenchsetup{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 (babel-french being an option of Babel, it cannot load a package while being read).

```
\newif\ifFB@poorman
\newdimen\FB@Mht
\ifLaTeXe
  \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 \frenchsetup{}.

```
\newcommand*{\FBsupR}{-0.12}
\newcommand*{\FBsupS}{0.65}
\newcommand*{\FB@lc}[1]{\MakeLowercase{#1}}
\DeclareRobustCommand*{\FB@up@fake}[1]{%
  \settoheight{\FB@Mht}{M}%
  \addtolength{\FB@Mht}{\FBsupR \FB@Mht}%
  \addtolength{\FB@Mht}{-\FBsupS ex}%
  \raisebox{\FB@Mht}{\scalefont{\FBsupS}{\FB@lc{#1}}}%
  }
```

The only packages I currently know to take advantage of real superscripts are a) realscripts used in conjunction with XeLaTeX 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.

```
\def\FB@split#1#2#3#4\@nil{\def\FB@firstthree{#1#2#3}%
                           \def\FB@suffix{#4}}
\def\FB@x{x}
\def\FB@j{j}
\DeclareRobustCommand*{\FB@up}[1]{%
  \bgroup \FB@poormantrue
    \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.

```
    \edef\reserved@a{\lowercase{%
       \noexpand\IfFileExists{\f@encoding\FB@firstthree -sup.fd}}}%
    \reserved@a
```

```
        {\ifx\FB@suffix\FB@x \FB@poormanfalse\fi
         \ifx\FB@suffix\FB@j \FB@poormanfalse\fi
         \ifFB@poorman \FB@up@fake{#1}%
         \else         \FB@up@real{#1}%
         \fi}%
        {\FB@up@fake{#1}}%
    \egroup}
```

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

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

\fup is defined as \FB@up unless \realsuperscript is defined by realscripts.sty. \fup just prints its argument in bookmarks.

```
  \DeclareRobustCommand*{\fup}[1]{%
    \texorpdfstring{\ifx\realsuperscript\@undefined
                      \FB@up{#1}%
                    \else
                      \bgroup\let\fakesuperscript\FB@up@fake
                        \realsuperscript{\FB@lc{#1}}\egroup
                    \fi
                    }{#1}%
  }
```

Poor man’s definition of \up for Plain.

```
\else
  \providecommand*{\up}[1]{\leavevmode\raise1ex\hbox{\sevenrm #1}}
\fi
```

\ieme
\ier
\iere
\iemes
\iers
\ieres

Some handy macros for those who don’t know how to abbreviate ordinals:

```
\def\ieme{\up{e}\xspace}
\def\iemes{\up{es}\xspace}
\def\ier{\up{er}\xspace}
\def\iers{\up{ers}\xspace}
\def\iere{\up{re}\xspace}
\def\ieres{\up{res}\xspace}
```

\FBmedkern
\FBthickkern

```
\newcommand*{\FBmedkern}{\kern+.2em}
\newcommand*{\FBthickkern}{\kern+.3em}
```

\primo
\fprimo)
\nos
\Nos
\No
\no

Some support macros relying on \up for numbering,

```
\newcommand*{\FrenchEnumerate}[1]{%
   #1\texorpdfstring{\up{o}\FBthickkern}{\textdegree\space}}
\newcommand*{\FrenchPopularEnumerate}[1]{%
   #1\texorpdfstring{\up{o})\FBthickkern}{\textdegree\space}}
```

Typing \primo should result in ‘^{o} ’ (except in bookmarks where \textdegree is used instead of o-superior),

```
\def\primo{\FrenchEnumerate1}
\def\secundo{\FrenchEnumerate2}
\def\tertio{\FrenchEnumerate3}
\def\quarto{\FrenchEnumerate4}
```

while typing \fprimo) gives ‘^{o}) (except in bookmarks where \textdegree is used instead),.

```
\def\fprimo){\FrenchPopularEnumerate1}
\def\fsecundo){\FrenchPopularEnumerate2}
\def\ftertio){\FrenchPopularEnumerate3}
\def\fquarto){\FrenchPopularEnumerate4}
```

Let’s provide four macros for the common abbreviations of “Numéro”. In bookmarks ° is used instead of o-superior.

```
\DeclareRobustCommand*{\No}{%
  \texorpdfstring{N\up{o}\FBmedkern}{N\textdegree\space}}
\DeclareRobustCommand*{\no}{%
    \texorpdfstring{n\up{o}\FBmedkern}{n\textdegree\space}}
\DeclareRobustCommand*{\Nos}{%
    \texorpdfstring{N\up{os}\FBmedkern}{N\textdegree\space}}
\DeclareRobustCommand*{\nos}{%
    \texorpdfstring{n\up{os}\FBmedkern}{n\textdegree\space}}
```

\bname \bsc These commands are meant to easily enter family names (in small capitals for the latter) while avoidind hyphenation. A \kern0pt is used instead of \mbox because \mbox would break microtype’s font expansion; as a positive side effect, composed names (such as Dupont-Durand) can now be hyphenated on explicit hyphens.

```
\ifLaTeXe
  \DeclareRobustCommand*{\bname}[1]{%
    \texorpdfstring{\leavevmode\begingroup\kern0pt #1\endgroup}{#1}%
  }
  \DeclareRobustCommand*{\bsc}[1]{%
    \texorpdfstring{\leavevmode\begingroup\kern0pt \scshape #1\endgroup}%
                   {\textsc{#1}}%
  }
\else
  \newcommand*{\bname}[1]{\leavevmode\begingroup\kern0pt #1\endgroup}
  \let\bsc\bname
\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.

```
\ifFBunicode
  \providecommand*{\textbackslash}{{\char"005C}}
  \providecommand*{\textasciicircum}{{\char"005E}}
  \providecommand*{\textasciitilde}{{\char"007E}}
  \DeclareRobustCommand*{\degre}{°}
\else
  \DeclareRobustCommand*{\degre}{\textdegree}
\fi
\DeclareRobustCommand*{\boi}{\textbackslash}
\DeclareRobustCommand*{\circonflexe}{\textasciicircum}
\DeclareRobustCommand*{\tild}{\textasciitilde}
\newcommand*{\at}{@}
```

\degres We now define a macro \degres for typesetting the abbreviation for 'degrees' (as in '°C' or '°K') in text fonts which also works in math mode for angles.

```
\DeclareRobustCommand*{\degres}{\degre}
\ifLaTeXe
  \AtBeginDocument{%
    \IfPackageLoadedTF{fontspec}{}%
        {\DeclareRobustCommand*{\degres}{%
          \texorpdfstring{\hbox{\UseTextSymbol{TS1}{\textdegree}}}%
                         {\textdegree}}%
        }%
  }
\fi
```

2.6 Formatting numbers

\StandardMathComma \DecimalMathComma As mentioned in the TEXbook p. 134, the comma is of type \mathpunct in math mode: it is automatically followed by a thin 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.

Unfortunately, \newcount inside \if breaks Plain formats.

```
\newif\ifFB@icomma
\newcount\mc@charclass
\newcount\mc@charfam
\newcount\mc@charslot
\newcount\std@mcc
\newcount\dec@mcc
\std@mcc=\mathcode`\,
\dec@mcc=\std@mcc
\@tempcnta=\std@mcc
\divide\@tempcnta by "1000
```

```
\multiply\@tempcnta by "1000
\advance\dec@mcc by -\@tempcnta
\newcommand*{\dec@math@comma}{\mathcode`\,=\dec@mcc}
\newcommand*{\std@math@comma}{\mathcode`\,=\std@mcc}
\let\dec@m@c\relax
```

If `\DecimalMathComma` is issued in the document body (when the current language is French) its effect will survive to a language switch, unless issued inside a group (see `\dec@m@c`'s expansion). The `icomma` inhibits `\DecimalMathComma`.

```
\newif\if@FBpreamble
\ifLaTeXe \@FBpreambletrue \fi
\newif\if@preamble@DecimalMathComma
\newcommand*{\DecimalMathComma}{%
  \if@FBpreamble \@preamble@DecimalMathCommatrue
  \else
    \ifFB@icomma
      \PackageWarning{french3.ldf}{%
        icomma package loaded, \protect\DecimalMathComma\MessageBreak
        does nothing.  Reported}%
    \else
      \ifFBfrench
        \dec@math@comma
        \let\dec@m@c\dec@math@comma
        \expandafter\addto\csname extras\languagename\endcsname
          {\dec@m@c}%
      \fi
    \fi
  \fi
}
\newcommand*{\StandardMathComma}{%
  \ifFB@icomma
    \PackageWarning{french3.ldf}{%
      icomma package loaded, \protect\StandardMathComma\MessageBreak
      does nothing.  Reported}%
  \else
    \ifFBfrench
      \std@math@comma
      \let\dec@m@c\relax
    \fi
  \fi
}
```

This is for Plain formats *only* (see below).

```
\ifLaTeXe\else
  \addto\noextrasfrench{\std@math@comma}
\fi
```

Fake command `\nombre` for Plain based formats, warning users of `babel-french` v. 1.x. about the change:

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

Cleanup and exit without loading any `.cfg` file in case of Plain formats.

```
\let\FBstop@here\relax
\def\FBclean@on@exit{%
  \let\ifLaTeXe\iffalse
  \let\LaTeXetrue\undefined
  \let\LaTeXefalse\undefined
  \let\FB@llc\loadlocalcfg
  \let\loadlocalcfg\@gobble}
\ifx\magnification\@undefined
\else
  \def\FBstop@here{%
    \FBclean@on@exit
    \ldf@finish\CurrentOption
    \let\loadlocalcfg\FB@llc
    \endinput}
\fi
\FBstop@here
```

What follows is for LaTeX2e *only*: the next piece of code would break Plain formats. If issued in the preamble, `\DecimalMathComma` works globally on all parts of the document that are typeset in a French dialect. Can be canceled anytime by `\StandardMathComma`.

```
\AddToHook{env/document/before}{%
  \@FBpreamblefalse
  \IfPackageLoadedTF{icomma}%
     {\FB@icommatrue
      \if@preamble@DecimalMathComma
        \PackageWarning{french3.ldf}{%
          icomma package loaded, \protect\DecimalMathComma%
          \MessageBreak does nothing.  Reported}%
      \fi
     }%
     {\if@preamble@DecimalMathComma
         \ifFB@mainlanguage@FR \dec@math@comma \fi
         \let\dec@m@c\dec@math@comma
         \addto\extrasfrench{\dec@m@c}%
      \fi
```

The comma is reset to type `\mathpunct` when leaving French dialects (only if the icomma package is not loaded).

```
      \addto\noextrasfrench{\std@math@comma}%
     }%
  }
```

nombre We redefine \nombre for LaTeX2e. The command \nombre is now borrowed from numprint.sty for LaTeX2e. There is no point to maintain the former tricky code when a package is dedicated to do the same job and more. 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 babel-french because of possible options conflict.

```
\renewcommand*{\nombre}[1]{\Warning@nombre{#1}}
\newcommand*{\Warning@nombre}[1]{%
   \ifdefined\numprint
     \numprint{#1}%
   \else
     \PackageWarning{french3.ldf}{%
        \protect\nombre\space now relies on package numprint.sty,%
        \MessageBreak add \protect
        \usepackage[autolanguage]{numprint},\MessageBreak
        see file numprint.pdf for more options.\MessageBreak
        \protect\nombre\space called}%
     \global\let\Warning@nombre\relax
     {#1}%
   \fi
}
```

```
\newcommand*{\FBthousandsep}{\kern \fontdimen2\font \relax}
```

2.7 Caption names

The next step consists in defining the French equivalents for the LaTeX caption names. New implementation for caption names (requires Babel's 3.10 or newer).

```
\StartBabelCommands*{french}{captions}
      [unicode, fontenc=TU EU1 EU2, charset=utf8]
   \SetString{\refname}{Références}
   \SetString{\abstractname}{Résumé}
   \SetString{\prefacename}{Préface}
   \SetString{\contentsname}{Table des matières}
   \SetString{\ccname}{Copie à }
   \SetString{\proofname}{Démonstration}
   \SetString{\partfirst}{Première}
   \SetString{\partsecond}{Deuxième}
   \SetStringLoop{ordinal#1}{%
     \frenchpartfirst,\frenchpartsecond,Troisième,Quatrième,%
     Cinquième,Sixième,Septième,Huitième,Neuvième,Dixième,Onzième,%
     Douzième,Treizième,Quatorzième,Quinzième,Seizième,%
     Dix-septième,Dix-huitième,Dix-neuvième,Vingtième}
\StartBabelCommands*{french}{captions}
   \SetString{\refname}{R\'ef\'erences}
   \SetString{\abstractname}{R\'esum\'e}
   \SetString{\bibname}{Bibliographie}
   \SetString{\prefacename}{Pr\'eface}
   \SetString{\chaptername}{Chapitre}
   \SetString{\appendixname}{Annexe}
   \SetString{\contentsname}{Table des mati\`eres}
   \SetString{\listfigurename}{Table des figures}
   \SetString{\listtablename}{Liste des tableaux}
   \SetString{\indexname}{Index}
   \SetString{\figurename}{Figure}
   \SetString{\tablename}{Table}
   \SetString{\pagename}{page}
   \SetString{\seename}{voir}
   \SetString{\alsoname}{voir aussi}
   \SetString{\enclname}{P.~J. }
   \SetString{\ccname}{Copie \`a }
   \SetString{\headtoname}{}
   \SetString{\proofname}{D\'emonstration}
   \SetString{\glossaryname}{Glossaire}
```

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

```
   \SetString{\partfirst}{Premi\`ere}
   \SetString{\partsecond}{Deuxi\`eme}
   \SetString{\partnameord}{partie}
   \SetStringLoop{ordinal#1}{%
     \partfirst,\partsecond,Troisi\`eme,Quatri\`eme, Cinqui\`eme,%
     Sixi\`eme,Septi\`eme,Huiti\`eme,Neuvi\`eme,Dixi\`eme,%
     Onzi\`eme,Douzi\`eme,Treizi\`eme,Quatorzi\`eme,Quinzi\`eme,%
     Seizi\`eme,Dix-septi\`eme,Dix-huiti\`eme,Dix-neuvi\`eme,%
     Vingti\`eme}
   \AfterBabelCommands{%
     \DeclareRobustCommand*{\FB@emptypart}{\def\thepart{\unskip}}%
     \DeclareRobustCommand*{\FB@partname}{%
        \ifFBPartNameFull
          \csname ordinal\romannumeral\value{part}\endcsname\space
          \partnameord\FB@emptypart
        \else
          Partie%
        \fi}%
    }
   \SetString{\partname}{\FB@partname}
\EndBabelCommands
```

\figurename and \tablename are printed in small caps in French, unless either SmallCapsFigTabCaptions is set to false or a class or package loaded before babel-french defines \FBfigtabshape as \relax.

```
\providecommand*{\FBfigtabshape}{\scshape}
```

2.8 Figure and table captions

\FBWarning \FBWarning is an alias of \PackageWarning{french3.ldf} which can be made silent by option SuppressWarning.

```
\newcommand{\FBWarning}[1]{\PackageWarning{french3.ldf}{#1}}
```

\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 LaTeX2e classes (a space should preceed the colon in French). This flaw may occur with pdfLaTeX as ':' is made active too late. With XeLaTeX, this glitch doesn't occur, you get 'Figure 1 : ' which is correct in French. With pdfLaTeX babel-french 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 LaTeX2e 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, babel-french just adds a hook called \FBCaption@Separator to \@makecaption; \FBCaption@Separator defaults to ': ' as in the standard definition of \@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 (babel-french makes ':' active and spanish.ldf makes '>' active).

```
\bgroup
  \catcode`: =12 \catcode`> =12 \relax
  \long\gdef\STD@makecaption#1#2{%
    \vskip\abovecaptionskip
    \sbox\@tempboxa{#1: #2}%
    \ifdim \wd\@tempboxa >\hsize
      #1: #2\par
    \else
      \global \@minipagefalse
      \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
    \fi
    \vskip\belowcaptionskip}
\egroup
```

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, babel-french 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.

Enable the standard warning only if high punctuation is active.

```
\newif\if@FBwarning@capsep
\ifFB@active@punct\@FBwarning@capseptrue\fi
\newcommand*{\CaptionSeparator}{\space\textendash\space}
\def\FBCaption@Separator{: }
\long\def\FB@makecaption#1#2{%
  \vskip\abovecaptionskip
  \sbox\@tempboxa{#1\FBCaption@Separator #2}%
  \ifdim \wd\@tempboxa >\hsize
    #1\FBCaption@Separator #2\par
  \else
    \global \@minipagefalse
    \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
  \fi
  \vskip\belowcaptionskip}
```

Disable the standard warning with AMS and SMF classes.

```
\@ifclassloaded{amsart}{\@FBwarning@capsepfalse}{}
\@ifclassloaded{amsbook}{\@FBwarning@capsepfalse}{}
\@ifclassloaded{amsdtx}{\@FBwarning@capsepfalse}{}
\@ifclassloaded{amsldoc}{\@FBwarning@capsepfalse}{}
\@ifclassloaded{amproc}{\@FBwarning@capsepfalse}{}
\@ifclassloaded{smfart}{\@FBwarning@capsepfalse}{}
\@ifclassloaded{smfbook}{\@FBwarning@capsepfalse}{}
```

Disable the standard warning for some classes that do not use ':' as caption separator.

```
\@ifclassloaded{IEEEconf}{\@FBwarning@capsepfalse}{}
\@ifclassloaded{IEEEtran}{\@FBwarning@capsepfalse}{}
\@ifclassloaded{revtex4-2}{\@FBwarning@capsepfalse}{}
\@ifclassloaded{svjour3}{\@FBwarning@capsepfalse}{}
```

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`)

```
\@ifclassloaded{memoir}{\@FBwarning@capsepfalse}{}
\ifFB@koma \@FBwarning@capsepfalse \fi
```

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

```
\@ifclassloaded{beamer}{\@FBwarning@capsepfalse}{}
\ifdefined\@makecaption\else\@FBwarning@capsepfalse\fi
```

Check the definition of `\@makecaption` (`\AtBeginDocument`, `caption3` compatibility), change it or issue a warning in case it has been changed by a class or package not (yet) compatible with babel-french; then change the definition of \FBCaption@Separator, taking care that the colon is typeset correctly in French (*not* ‘Figure 1: légende’).

```
\AtBeginDocument{%
  \ifx\@makecaption\STD@makecaption
    \global\let\@makecaption\FB@makecaption
```

If OldFigTabCaptions=true, do not overwrite \FBCaption@Separator (already saved as ‘: ’ for other languages and set to \CaptionSeparator by \extrasfrench when French is the main language); otherwise locally force \autospace@beforeFDP in case AutoSpacePunctuation=false.

```
    \ifFBOldFigTabCaptions
    \else
      \def\FBCaption@Separator{{\autospace@beforeFDP : }}%
      \ifFBCustomiseFigTabCaptions
        \ifFB@mainlanguage@FR
          \def\FBCaption@Separator{\CaptionSeparator}%
        \fi
      \fi
    \fi
    \@FBwarning@capsepfalse
  \fi
```

No Warning if caption.sty or caption-light.sty has been loaded.

```
    \IfPackageLoadedTF{caption}{\@FBwarning@capsepfalse}{}%
    \IfPackageLoadedTF{caption-light}{\@FBwarning@capsepfalse}{}%
```

Final warning if relevant:

```
  \if@FBwarning@capsep
     \FBWarning
       {Figures' and tables' captions might look like\MessageBreak
        `Figure 1:' in French instead of `Figure 1 :'.\MessageBreak
        If this happens, to fix this issue\MessageBreak
        switch to LuaLaTeX or XeLaTeX or\MessageBreak
        try to add \protect\usepackage{caption} or\MessageBreak
        ... leave it as it is; reported}%
  \fi
  \let\FB@makecaption\relax
  \let\STD@makecaption\relax
}
```

2.9 Dots...

\FBtextellipsis Unless a ready-made character is available in the current font, LaTeX’s default definition of \textellipsis includes a \kern at the end; this space is not wanted in some cases (before a closing brace for instance) and \kern breaks hyphenation of the next word. We define \FBtextellipsis for French (in LaTeX only) the same way but without the last \kern.

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

```
\ifFBunicode
\else
  \DeclareTextSymbol{\FBtextellipsis}{LY1}{133}
  \DeclareTextCommand{\FBtextellipsis}{PU}{\9040\046}
  \DeclareTextCommand{\FBtextellipsis}{PD1}{\203}
  \DeclareTextCommandDefault{\FBtextellipsis}{%
      .\kern\fontdimen3\font.\kern\fontdimen3\font.\xspace}%
  \def\bbl@frenchdots{\babel@save\textellipsis
                        \let\textellipsis\FBtextellipsis}
  \addto\extrasfrench{\bbl@frenchdots}
\fi
```

2.10 More checks about packages' loading order

Like packages `captions` and `floatrow` (see section 2.8), package `listings` should be loaded after `babel-french` due to active characters issues (pdfLaTeX only).

```
\ifFB@active@punct
  \IfPackageLoadedTF{listings}
    {\AddToHook{env/document/before}{%
       \FBWarning{Please load the "listings" package\MessageBreak
                  AFTER babel/french; reported}}%
    }{}
\fi
```

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

```
\newif\if@FBwarning@natbib
\ifFB@active@punct
  \IfPackageLoadedTF{natbib}{}{\@FBwarning@natbibtrue}
\fi
\AddToHook{env/document/before}{%
  \if@FBwarning@natbib
    \IfPackageLoadedTF{natbib}{}{\@FBwarning@natbibfalse}%
  \fi
  \if@FBwarning@natbib
    \FBWarning{Please load the "natbib" package\MessageBreak
               BEFORE babel/french; reported}%
  \fi
}
```

Package `beamerarticle` should be loaded before `babel-french` to avoid list's conflicts, see p. 45.

```
\newif\if@FBwarning@beamerarticle
```

```
\IfPackageLoadedTF{beamerarticle}{}{\@FBwarning@beamerarticletrue}
\AddToHook{env/document/before}{%
   \if@FBwarning@beamerarticle
     \IfPackageLoadedTF{beamerarticle}{}%
                                      {\@FBwarning@beamerarticlefalse}%
   \fi
   \if@FBwarning@beamerarticle
     \FBWarning{Please load the "beamerarticle" package\MessageBreak
                BEFORE babel/french; reported}%
   \fi
}
```

2.11 Setup options: key/value stuff (l3keys)

Check LaTeX2e version (support for l3keys required).

```
\NeedsTeXFormat{LaTeX2e}[2022-06-01]
```

If the new templates for lists and footnotes are available, babel-french will use them.

```
\newif\ifFBnewlists
\newif\ifFBnewfootnotes
\IfPackageLoadedTF{latex-lab-testphase-block}{\FBnewliststrue}{}
\IfPackageLoadedTF{latex-lab-testphase-block}{\FBnewfootnotestrue}{}
```

All setup options are handled by command \frenchsetup{} based on the l3keys' \SetKeys{} command. A list of flags is defined beforehand and set to default values which will possibly be changed 'AtEndOfPackage' in case French is the main language. After this, \frenchsetup{} eventually modifies the preset values of these flags.
Some options processing occurs in \frenchsetup{}, *only for options explicitly set* by \frenchsetup{}, the rest 'AtBeginDocument'; any option affecting \extrasfrench{} *must* be immediatelt processed by \frenchsetup{}: 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 babel-french 'AtBeginDocument'. Reexecuting \extrasfrench{} is not an option because of its side-effects (f.i. \babel@save and \babel@savevariable did not work for French).
We first define a collection of conditionals and set their defaults (true or false).

```
\newif\ifFBShowOptions
\newif\ifFBStandardLayout            \FBStandardLayouttrue
\newif\ifFBStandardListSpacing       \FBStandardListSpacingtrue
\newif\ifFBListOldLayout
\newif\ifFBListItemsAsPar
\newif\ifFBCompactItemize
\newif\ifFBStandardItemizeEnv        \FBStandardItemizeEnvtrue
\newif\ifFBStandardEnumerateEnv      \FBStandardEnumerateEnvtrue
\newif\ifFBStandardItemLabels        \FBStandardItemLabelstrue
\newif\ifFBStandardLists             \FBStandardListstrue
\newif\ifFBIndentFirst
```

```
\newif\ifFBFrenchFootnotes
\newif\ifFBAutoSpaceFootnotes
\newif\ifFBOriginalTypewriter
\newif\ifFBThinColonSpace
\newif\ifFBThinSpaceInFrenchNumbers
\newif\ifFBFrenchSuperscripts       \FBFrenchSuperscriptstrue
\newif\ifFBLowercaseSuperscripts    \FBLowercaseSuperscriptstrue
\newif\ifFBPartNameFull             \FBPartNameFulltrue
\newif\ifFBCustomiseFigTabCaptions
\newif\ifFBOldFigTabCaptions
\newif\ifFBSmallCapsFigTabCaptions  \FBSmallCapsFigTabCaptionstrue
\newif\ifFBSuppressWarning
\newif\ifFBINGuillSpace
```

The following patch is for koma-script classes: the `\partformat` command, defined as `\partname~\thepart\autodot`, is incompatible with our redefinition of `\partname`.

```
\ifFB@koma
  \ifdefined\partformat
    \def\FB@partformat@fix{%
           \ifFBPartNameFull
             \babel@save\partformat
             \renewcommand*{\partformat}{\partname}%
           \fi}
    \addto\extrasfrench{\FB@partformat@fix}%
  \fi
\fi
```

The defaults values of these flags are choosen so that `babel-french` does not change anything regarding the global layout. Some of them must be toogled when French (or a French dialect) is the main language. The latter (last option of `Babel`, stored in `\bbl@main@language`) will be known ‘AtEndOfPackage’. So we postpone the `\bbl@main@language` checking until then.

Our list customisation conflicts with the `beamer` class and with the `beamerarticle` package. The patch provided in `beamerbasecompatibility` solves the conflict except in case of language changes, so we provide our own patch. When the `beamer` is loaded, lists are not customised at all to ensure compatibility. The `beamerarticle` package needs to be loaded *before* `Babel`, a warning is issued otherwise, see section 2.10; a light customisation is compatible with the `beamerarticle` package.

```
\def\FB@french{french}
\newif\ifFB@mainlanguage@FR
\AtEndOfPackage{%
  \ifx\bbl@main@language\FB@french \FB@mainlanguage@FRtrue \fi
  \ifFB@mainlanguage@FR
    \@ifclassloaded{beamer}%
      {\PackageInfo{french3.ldf}{%
          No list customisation for the beamer class,%
          \MessageBreak reported}}%
      {\IfPackageLoadedTF{beamerarticle}%
         {\FBStandardItemLabelsfalse
          \FBStandardListSpacingfalse
          \PackageInfo{french3.ldf}{%
             Minimal list customisation for the beamerarticle%
             \MessageBreak package; reported}}%
```

Otherwise customise lists “à la française”:

```
         {\FBStandardListSpacingfalse
          \FBStandardItemizeEnvfalse
          \FBStandardEnumerateEnvfalse
          \FBStandardItemLabelsfalse}%
      }
    \FBIndentFirsttrue
    \FBFrenchFootnotestrue
    \FBAutoSpaceFootnotestrue
    \FBCustomiseFigTabCaptionstrue
  \fi
}
```

\frenchsetup Let’s define the keys to be used in \frenchsetup{}.

```
\DeclareKeys[FBsetup]
  {
    ShowOptions.if               = FBShowOptions                        ,
    StandardLayout.default:n     = {true}                               ,
    StandardLayout.code          = \FBStandardLayout@setup{#1}          ,
    StandardListSpacing.if       = FBStandardListSpacing                ,
    ReduceListSpacing.ifnot      = FBStandardListSpacing                ,
    ListOldLayout.default:n      = {true}                               ,
    ListOldLayout.code           = \FBListOldLayout@setup{#1}           ,
    CompactItemize.default:n     = {true}                               ,
    CompactItemize.code          = \FBCompactItemize@setup{#1}          ,
    StandardItemizeEnv.if        = FBStandardItemizeEnv                 ,
    StandardEnumerateEnv.if      = FBStandardEnumerateEnv               ,
    StandardItemLabels.if        = FBStandardItemLabels                 ,
    ItemLabels.store             = \FrenchLabelItem                     ,
    ItemLabeli.store             = \Frlabelitemi                        ,
    ItemLabelii.store            = \Frlabelitemii                       ,
    ItemLabeliii.store           = \Frlabelitemiii                      ,
    ItemLabeliv.store            = \Frlabelitemiv                       ,
    StandardLists.default:n      = {true}                               ,
    StandardLists.code           = \FBStandardLists@setup{#1}           ,
    ListItemsAsPar.if            = FBListItemsAsPar                     ,
    IndentFirst.if               = FBIndentFirst                        ,
    FrenchFootnotes.if           = FBFrenchFootnotes                    ,
    AutoSpaceFootnotes.if        = FBAutoSpaceFootnotes                 ,
    AutoSpacePunctuation.if      = FBAutoSpacePunctuation               ,
    OriginalTypewriter.if        = FBOriginalTypewriter                 ,
    ThinColonSpace.default:n     = {true}                               ,
    ThinColonSpace.code          = \FBThinColonSpace@setup{#1}          ,
    ThinSpaceInFrenchNumbers.if  = FBThinSpaceInFrenchNumbers           ,
    FrenchSuperscripts.if        = FBFrenchSuperscripts                 ,
    LowercaseSuperscripts.if     = FBLowercaseSuperscripts              ,
    PartNameFull.if              = FBPartNameFull                       ,
    CustomiseFigTabCaptions.if   = FBCustomiseFigTabCaptions            ,
    OldFigTabCaptions.default:n  = {true}                               ,
    OldFigTabCaptions.code       = \FBOldFigTabCaptions@setup{#1}       ,
    SmallCapsFigTabCaptions.default:n = {true}                          ,
    SmallCapsFigTabCaptions.code = \FBSmallCapsFigTabCaptions@setup{#1} ,
    SuppressWarning.default:n    = {true}                               ,
    SuppressWarning.code         = \FBSuppressWarning@setup{#1}         ,
    INGuillSpace.default:n       = {true}                               ,
    INGuillSpace.code            = \FBINGuillSpace@setup{#1}            ,
    InnerGuillSingle.if          = FBInnerGuillSingle                   ,
    EveryParGuill.default:n      = {open}                               ,
    EveryParGuill.code           = \FBEveryParGuill@setup{#1}           ,
    og.code                      = \FBog@setup{#1}                      ,
    fg.code                      = \FBfg@setup{#1}                      ,
}
```

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`. `\frenchsetup{}` can only be called in the preamble.

```
\newcommand*{\frenchsetup}[1]{%
  \SetKeys[FBsetup]{#1}%
}%
\@onlypreamble\frenchsetup
```

Keep the former name `\frenchbsetup` working for compatibility.

```
\let\frenchbsetup\frenchsetup
\@onlypreamble\frenchbsetup
```

The following commands, defined with property `.code` in `DeclareKeys{}`, execute some post-treatment required to immediately take the flags value into account.

```
\newcommand*{\FBStandardLayout@setup}[1]%
  {\ifFB@mainlanguage@FR
     \csname FBStandardLayout#1\endcsname
   \else
     \PackageWarning{french3.ldf}%
         {Option `StandardLayout' skipped:\MessageBreak
          French is *not* babel's last option.\MessageBreak
          Reported}%
   \fi
   \ifFBStandardLayout
```

```
     \FBStandardListSpacingtrue
     \FBStandardItemizeEnvtrue
     \FBStandardItemLabelstrue
     \FBStandardEnumerateEnvtrue
     \FBIndentFirstfalse
     \FBFrenchFootnotesfalse
     \FBAutoSpaceFootnotesfalse
   \else
     \FBStandardListSpacingfalse
     \FBStandardItemizeEnvfalse
     \FBStandardItemLabelsfalse
     \FBStandardEnumerateEnvfalse
     \FBIndentFirsttrue
     \FBFrenchFootnotestrue
     \FBAutoSpaceFootnotestrue
   \fi
  }
\newcommand*{\FBListOldLayout@setup}[1]%
  {\csname FBListOldLayout#1\endcsname
   \ifFBListOldLayout
     \FBStandardEnumerateEnvtrue
     \renewcommand*{\FrenchLabelItem}{\textendash}%
   \fi
  }
\newcommand*{\FBCompactItemize@setup}[1]%
  {\csname FBCompactItemize#1\endcsname
   \ifFBCompactItemize
     \FBStandardItemizeEnvfalse
     \FBStandardEnumerateEnvfalse
   \else
     \FBStandardItemizeEnvtrue
     \FBStandardEnumerateEnvtrue
   \fi
  }
\newcommand*{\FBStandardLists@setup}[1]%
  {\csname FBStandardLists#1\endcsname
   \ifFBStandardLists
     \FBStandardListSpacingtrue
     \FBStandardItemizeEnvtrue
     \FBStandardEnumerateEnvtrue
     \FBStandardItemLabelstrue
   \else
     \FBStandardListSpacingfalse
     \FBStandardItemizeEnvfalse
     \FBStandardEnumerateEnvfalse
     \FBStandardItemLabelsfalse
   \fi
  }
\newcommand*{\FBThinColonSpace@setup}[1]%
  {\csname FBThinColonSpace#1\endcsname
   \ifFBThinColonSpace
     \renewcommand*{\FBcolonspace}{\FBthinspace}%
   \fi
  }
\newcommand*{\FBOldFigTabCaptions@setup}[1]%
  {\csname FBOldFigTabCaptions#1\endcsname
   \ifFBOldFigTabCaptions
     \def\FB@capsep@fix{\babel@save\FBCaption@Separator
             \def\FBCaption@Separator{\CaptionSeparator}}%
             \addto\extrasfrench{\FB@capsep@fix}%
   \fi
  }
\newcommand*{\FBSmallCapsFigTabCaptions@setup}[1]%
  {\csname FBSmallCapsFigTabCaptions#1\endcsname
   \ifFBSmallCapsFigTabCaptions
   \else
     \let\FBfigtabshape\relax
   \fi
  }
\newcommand*{\FBSuppressWarning@setup}[1]%
  {\csname FBSuppressWarning#1\endcsname
   \ifFBSuppressWarning
     \renewcommand{\FBWarning}[1]{}%
   \fi
  }
\newcommand*{\FBINGuillSpace@setup}[1]%
  {\csname FBINGuillSpace#1\endcsname
   \ifFBINGuillSpace
     \renewcommand*{\FBguillspace}{\space}%
   \fi
  }
\newcommand*{\FBEveryParGuill@setup}[1]%
  {\expandafter\let\expandafter
      \FBeveryparguill\csname FBguill#1\endcsname
   \ifx\FBeveryparguill\FBguillopen
   \else\ifx\FBeveryparguill\FBguillclose
        \else\ifx\FBeveryparguill\FBguillnone
             \else
               \let\FBeveryparguill\FBguillopen
               \FBWarning{Wrong value for `EveryParGuill':
                          try `open',\MessageBreak
                          `close' or `none'. Reported}%
```

```
              \fi
        \fi
   \fi
  }
```

Inputing French quotes as *single characters* when they are available on the keyboard (through a compose key for instance) is more comfortable than typing \og and \fg. Life is simple here with modern XeTeX engines: we just have to set \XeTeXcharclass of quotes to the proper value for XeTeX.

With pdfTeX (or old 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 non-breaking 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 next command is meant for checking whether a character is single-byte (\FB@second is empty) or not.

```
\def\FB@parse#1#2\endparse{\def\FB@second{#2}}%
\newcommand*{\FB@@og}%
  {\ifFBfrench
     \ifFB@spacing \FB@og\ignorespaces
     \else \guillemotleft
     \fi
   \else \guillemotleft
   \fi
  }
\newcommand*{\FB@@fg}%
  {\ifFBfrench
     \ifFB@spacing \FB@fg
     \else \guillemotright
     \fi
   \else \guillemotright
   \fi
  }
\newcommand*{\FBog@setup}[1]%
  {\ifFBunicode
```

With the XeTeX engine, French guillemets will have their \XeTeXcharclass(es) set in \extrasfrench according to this flag.

```
     \FB@og@fg@xetextrue
   \else
```

This is for conventional TeX engines:

```
     \AddToHook{env/document/before}{%
        \ifdefined\uc@dclc
```

Package inputenc with utf8x (ucs) encoding loaded, use \uc@dclc:

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

if encoding is not utf8x, check if the argument of og is a single-byte character:

```
          \FB@parse#1\endparse
          \ifx\FB@second\@empty
```

This means 8-bit character encoding. Package MULEenc (from CJK) defines \mule@def to map characters to control sequences.

```
            \ifdefined\mule@def
              \mule@def{11}{\FB@@og}%
            \else
              \ifdefined\DeclareInputText
                \@tempcnta`#1\relax
                \DeclareInputText{\the\@tempcnta}{\FB@@og}%
              \else
```

Package inputenc not loaded, no way…

```
                \FBWarning{Option `og' requires package
                           inputenc;\MessageBreak reported}%
              \fi
            \fi
          \else
```

This means multi-byte character encoding, we assume UTF-8

```
           \DeclareUnicodeCharacter{00AB}{\FB@@og}%
          \fi
        \fi}%
   \fi
  }
```

Same code for the closing quote.

```
\newcommand*{\FBfg@setup}[1]%
  {\ifFBunicode
     \FB@og@fg@xetextrue
   \else
     \AddToHook{env/document/before}{%
        \ifdefined\uc@dclc
          \uc@dclc{187}{default}{\FB@@fg}%
        \else
          \FB@parse#1\endparse
          \ifx\FB@second\@empty
             \ifdefined\mule@def
               \mule@def{27}{{\FB@@fg}}%
             \else
               \ifdefined\DeclareInputText
                 \@tempcnta`#1\relax
                 \DeclareInputText{\the\@tempcnta}{\FB@@fg}%
               \else
```

```
                 \FBWarning{Option `fg' requires package
                            inputenc;\MessageBreak reported}%
                 \fi
               \fi
          \else
            \DeclareUnicodeCharacter{00BB}{\FB@@fg}%
          \fi
        \fi}%
   \fi
  }
% \end{macro}
%
% \begin{macro}{\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 \pkg{enumitem},
%    \pkg{paralist} and \pkg{enumerate}; then it processes the options
%    as set by \fbsetup{} or forced for compatibility with packages
%    loaded in the preamble.
%
%    When French is the main language, |\extrasfrench| and
%    |\captionsfrench| are executed by \babel{} at |\begin{document}|,
%    i.e.\ after |\FBprocess@options|.
%    \begin{macrocode}
\newcommand*{\FBprocess@options}{%
```

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

```
  \IfPackageLoadedTF{enumitem}{%
     \ifFBStandardItemizeEnv
     \else
       \FBStandardItemizeEnvtrue
       \PackageInfo{french3.ldf}%
          {Setting StandardItemizeEnv=true for\MessageBreak
           compatibility with enumitem package,\MessageBreak
           reported}%
     \fi
     \ifFBStandardEnumerateEnv
     \else
       \FBStandardEnumerateEnvtrue
       \PackageInfo{french3.ldf}%
          {Setting StandardEnumerateEnv=true for\MessageBreak
           compatibility with enumitem package,\MessageBreak
           reported}%
     \fi}{}%
  \IfPackageLoadedTF{paralist}{%
     \ifFBStandardItemizeEnv
```

```
   \else
     \FBStandardItemizeEnvtrue
     \PackageInfo{french3.ldf}%
        {Setting StandardItemizeEnv=true for\MessageBreak
         compatibility with paralist package,\MessageBreak
         reported}%
   \fi
   \ifFBStandardEnumerateEnv
   \else
     \FBStandardEnumerateEnvtrue
     \PackageInfo{french3.ldf}%
        {Setting StandardEnumerateEnv=true for\MessageBreak
         compatibility with paralist package,\MessageBreak
         reported}%
   \fi}{}%
\IfPackageLoadedTF{enumerate}{%
   \ifFBStandardEnumerateEnv
   \else
     \FBStandardEnumerateEnvtrue
     \PackageInfo{french3.ldf}%
        {Setting StandardEnumerateEnv=true for\MessageBreak
         compatibility with enumerate package,\MessageBreak
         reported}%
   \fi}{}%
\ifFB@mainlanguage@FR
\else
  \ifFBStandardItemizeEnv
  \else
    \PackageWarning{french3.ldf}%
      {babel-french will not customise lists' layout\MessageBreak
       when French is not the main language,\MessageBreak
       reported}%
  \fi
\fi
```

When tagging is enabled, lists’s customisation is disabled, a warning is issued. Legacy lists will be customised in `\extrafrench{}` which be called by babel later on (`\AtBeginDocument{}`).

```
\ifFBnewlists
  \FBWarning{You requested LaTeX tagging support.\MessageBreak
    Babel-french’s list customization is currently\MessageBreak
    incompatible with the new lists’ implementation\MessageBreak
    (still experimental) required to support tagging.%
    \MessageBreak Babel-french’s list customization is *DISABLED*%
    \MessageBreak when tagging is enabled (see frenchb.pdf).%
    \MessageBreak Reported
    }%
  \setlistindentFB
\fi
```

Options FrenchFootnotes and Option AutoSpaceFootnotes are handled here when new footnotes templates are available.

```
\ifFBnewfootnotes
  \ifdim\parindentFFN<\maxdimen
  \else
     \parindentFFN=\parindent
     \ifdim\parindentFFN<1.8em \parindentFFN=1.8em \fi
  \fi
  \settowidth{\FBfnindent}{\dotFFN\kernFFN}%
  \addtolength{\FBfnindent}{\parindentFFN}%
  \ifFBFrenchFootnotes
    \NewSocketPlug{fntext/mark}{french}
      {\hb@xt@ \footnotemargin{\hss\newfootnotemarkFB}}
    \AssignSocketPlug{fntext/mark}{french}
    \AddToHook{cmd/maketitle/before}
       {\AssignSocketPlug{fntext/mark}{default}}
    \AddToHook{cmd/maketitle/after}
       {\AssignSocketPlug{fntext/mark}{french}}
    \AddToHook{env/minipage/begin}
       {\AssignSocketPlug{fntext/mark}{default}}
    \AddToHook{fntext/para}{\parindent=\parindentFFN}
    \AddToHook{fntext/para}{\localleftbox{}}
    \AddToHook{fntext/para}{\let\FBeverypar@quote\relax}
  \fi
  \ifFBAutoSpaceFootnotes
    \AddToHook{fnmark/before}{\FBfnmarkspace}
  \fi
\fi
```

Option SmallCapsFigTabCaptions: \figurename and \tablename are printed in small caps (in French *only*), unless either SmallCapsFigTabCaptions is set to false or a class or package loaded defines \FBfigtabshape as \relax. As \figurename and \tablename should not include font commmands, we use \fnum@figure and \fnum@table when available (not in beamer.cls f.i.).

```
\ifx\FBfigtabshape\relax
\else
  \ifdefined\fnum@figure
    \let\fnum@figureORI\fnum@figure
    \renewcommand{\fnum@figure}{{\ifFBfrench\FBfigtabshape\fi
                                 \fnum@figureORI}}%
  \fi
  \ifdefined\fnum@table
    \let\fnum@tableORI\fnum@table
    \renewcommand{\fnum@table}{{\ifFBfrench\FBfigtabshape\fi
```

```
                                \fnum@tableORI}}%
  \fi
\fi
```

AutoSpacePunctuation, when true, adds a non-breaking space (in French only) before the four characters (:;!?) if and only if spacing is required by French typographic rules. When false, these characters are left unchanged.

```
\ifFBAutoSpacePunctuation
  \autospace@beforeFDP
\else
  \noautospace@beforeFDP
\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.

```
\ifFBOriginalTypewriter
\else
  \let\ttfamilyORI\ttfamily
  \let\rmfamilyORI\rmfamily
  \let\sffamilyORI\sffamily
  \let\ttfamily\ttfamilyFB
  \let\rmfamily\rmfamilyFB
  \let\sffamily\sffamilyFB
\fi
```

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

```
\IfPackageLoadedTF{numprint}%
  {\ifnprt@autolanguage
     \providecommand*{\npstylefrench}{}%
     \ifFBThinSpaceInFrenchNumbers
       \renewcommand*{\FBthousandsep}{\FBthinspace}%
     \fi
     \g@addto@macro\npstylefrench{\npthousandsep{\FBthousandsep}}%
   \fi
  }{}%
```

FrenchSuperscripts: if true \up=\fup, else \up=\textsuperscript. The star-form \up*=\FB@up@fake is provided for fonts that lack some superior letters: Adobe Jenson Pro and Utopia Expert have no "g superior" for instance.

```
\ifFBFrenchSuperscripts
  \DeclareRobustCommand*{\up}{%
    \texorpdfstring{\@ifstar{\FB@up@fake}{\fup}}{}%
    }
```

```
\else
  \DeclareRobustCommand*{\up}{%
    \texorpdfstring{\@ifstar{\FB@up@fake}{\textsuperscript}}{}%
    }
\fi
```

LowercaseSuperscripts: if false \FB@lc is redefined to do nothing.

```
\ifFBLowercaseSuperscripts
\else
  \renewcommand*{\FB@lc}[1]{##1}%
\fi
```

This is for koma-script, memoir and beamer classes. If the caption delimiter has been user customised, leave it unchanged. Otherwise, force the colon to behave properly in French (add locally \autospace@beforeFDP in case of AutoSpacePunctuation=false) and change the caption delimiter to \CaptionSeparator if CustomiseFigTabCaptions has been set to true.

```
\ifFB@koma
   \ifx\captionformat\FB@std@capsep
     \ifFBCustomiseFigTabCaptions
        \renewcommand*{\captionformat}{\CaptionSeparator}%
     \else
        \renewcommand*{\captionformat}{{\autospace@beforeFDP :\ }}%
     \fi
   \fi
\fi
\@ifclassloaded{memoir}%
   {\ifx\@contdelim\FB@std@capsep
      \ifFBCustomiseFigTabCaptions
        \captiondelim{\CaptionSeparator}%
      \else
        \captiondelim{{\autospace@beforeFDP : }}%
      \fi
    \fi}{}%
\@ifclassloaded{beamer}%
   {\protected@edef\FB@capsep{%
       \csname beamer@@tmpl@caption label separator\endcsname}%
    \ifx\FB@capsep\FB@std@capsep
      \ifFBCustomiseFigTabCaptions
         \defbeamertemplate{caption label separator}{FBcustom}{%
             \CaptionSeparator}%
         \setbeamertemplate{caption label separator}[FBcustom]%
      \else
         \defbeamertemplate{caption label separator}{FBcolon}{%
             {\autospace@beforeFDP : }}%
         \setbeamertemplate{caption label separator}[FBcolon]%
      \fi
    \fi}{}%
```

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

```
\ifFBShowOptions
  \GenericWarning{* }{%
   *** List of possible options for babel-french ***\MessageBreak
   [Default values between brackets when french is loaded *LAST*]%
   \MessageBreak
   ShowOptions [false]\MessageBreak
   StandardLayout [false]\MessageBreak
   PartNameFull [true]\MessageBreak
   IndentFirst [true]\MessageBreak
   ListItemsAsPar [false]\MessageBreak
   StandardListSpacing [false]\MessageBreak
   StandardItemizeEnv [false]\MessageBreak
   StandardEnumerateEnv [false]\MessageBreak
   StandardItemLabels [false]\MessageBreak
   ItemLabels=\textemdash, \textbullet,
      \protect\ding{43},... [\textendash]\MessageBreak
   ItemLabeli=\textemdash, \textbullet,
      \protect\ding{43},... [\textendash]\MessageBreak
   ItemLabelii=\textemdash, \textbullet,
      \protect\ding{43},... [\textendash]\MessageBreak
   ItemLabeliii=\textemdash, \textbullet,
      \protect\ding{43},... [\textendash]\MessageBreak
   ItemLabeliv=\textemdash, \textbullet,
      \protect\ding{43},... [\textendash]\MessageBreak
   StandardLists [false]\MessageBreak
   ListOldLayout [false]\MessageBreak
   FrenchFootnotes [true]\MessageBreak
   AutoSpaceFootnotes [true]\MessageBreak
   AutoSpacePunctuation [true]\MessageBreak
   ThinColonSpace [false]\MessageBreak
   OriginalTypewriter [false]\MessageBreak
   og= <left quote character>, fg= <right quote character>%
   INGuillSpace [false]\MessageBreak
   EveryParGuill=open, close, none [open]\MessageBreak
   InnerGuillSingle [false]\MessageBreak
   ThinSpaceInFrenchNumbers [false]\MessageBreak
   SmallCapsFigTabCaptions [true]\MessageBreak
   CustomiseFigTabCaptions [true]\MessageBreak
   OldFigTabCaptions [false]\MessageBreak
   FrenchSuperscripts [true]\MessageBreak
   LowercaseSuperscripts [true]\MessageBreak
   SuppressWarning [false]\MessageBreak
   \MessageBreak
   *********************************************%
```

```
     \MessageBreak\protect\frenchsetup{ShowOptions}}
  \fi
}
```

Just before `\begin{document}`, let's now process the remaining options, either not explicitly set by `\frenchsetup{}` or possibly modified by packages loaded after `babel-french`. We also have to provide an `\xspace` command in case the `xspace` package is not loaded.

```
\AddToHook{env/document/before}{%
   \providecommand*{\xspace}{\relax}%
   \FBprocess@options
```

Finally, with pdfLaTeX, when OT1 encoding is in use at the `\begin{document}` a warning is issued; `\encodingdefault` being defined as 'long', the test would fail if `\FBOTone` was defined with `\newcommand*`!

```
   \begingroup
     \newcommand{\FBOTone}{OT1}%
     \ifx\encodingdefault\FBOTone
       \FBWarning{OT1 encoding should not be used for French.%
                  \MessageBreak
                  Add \protect\usepackage[T1]{fontenc} to the
                  preamble\MessageBreak of your document; reported}%
     \fi
   \endgroup
}
```

2.12 French lists

`\listFB` `\listORI` `\FB@listVsettings`

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 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.

```
\let\listORI\list
\let\endlistORI\endlist
\newdimen\FB@pardim
\def\FB@listVsettings{%
      \setlength{\topsep}{0.8ex plus 0.4ex minus 0.4ex}%
      \setlength{\partopsep}{0.4ex plus 0.2ex minus 0.2ex}%
      \setlength{\itemsep}{0.4ex plus 0.2ex minus 0.2ex}%
      \setlength{\parsep}{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 \FB@pardim.

```
      \FB@pardim=\parskip
```

If \parskip is not null, \parsep is set to \parskip, so paragraphs inside items will be preceeded by the same vertical space as paragraphs located outside lists; the vertical skip before items (\itemsep + \parsep) doesn't need to be enlarged.

```
      \ifdim\FB@pardim>\z@
        \addtolength{\topsep}{-\FB@pardim}%
        \addtolength{\partopsep}{\FB@pardim}%
        \setlength{\parsep}{\FB@pardim}%
        \addtolength{\itemsep}{-\FB@pardim}%
      \fi
}
\def\listFB#1#2{\listORI{#1}{\FB@listVsettings #2}}
\let\endlistFB\endlistORI
```

Let's now consider French itemize-lists. They differ from those provided by the standard LaTeX classes:

- The '•' is never used in French itemize-lists, an emdash '—' or an endash '–' is preferred for all levels. The item label to be used in French, stored in \FrenchLabelItem}, defaults to '—' and can be changed using \frenchsetup{} (see section 2.11).
- 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 shown p. 6.

\FrenchLabelItem
\Frlabelitemi
\Frlabelitemii
\Frlabelitemiii
\Frlabelitemiv

Default labels for French itemize-lists —same label for all levels—, (already defined as empty by \DeclareKey{}):

```
\renewcommand*{\FrenchLabelItem}{\textemdash}
\renewcommand*{\Frlabelitemi}{\FrenchLabelItem}
\renewcommand*{\Frlabelitemii}{\FrenchLabelItem}
\renewcommand*{\Frlabelitemiii}{\FrenchLabelItem}
\renewcommand*{\Frlabelitemiv}{\FrenchLabelItem}
```

\listindentFB
\descindentFB
\labelindentFB
\labelwidthFB

Let's define four dimens \listindentFB, \descindentFB, \labelindentFB and \labelwidthFB to customise lists' horizontal indentations. They are given silly negative values here in order to eventually enable their customisation in the preamble. They will get reasonnable defaults later when entering French (see below

\setlistindentFB and \setlabelitemsFB) unless they have been customised before.

```
\newdimen\listindentFB
\setlength{\listindentFB}{-1pt}
\newdimen\descindentFB
\setlength{\descindentFB}{-1pt}
\newdimen\labelindentFB
\setlength{\labelindentFB}{-1pt}
\newdimen\labelwidthFB
\setlength{\labelwidthFB}{-1pt}
```

The next function will be included in \update@frenchlists which is exectuted in \extrasfrench{} 'AtBeginDocument'.

```
\def\setlistindentFB{%
  \ifdim\labelindentFB<\z@
    \ifdim\parindent=\z@
      \setlength{\labelindentFB}{1.5em}%
    \else
      \setlength{\labelindentFB}{\parindent}%
    \fi
  \fi
  \ifdim\listindentFB<\z@
    \ifdim\parindent=\z@
      \setlength{\listindentFB}{1.5em}%
    \else
      \setlength{\listindentFB}{\parindent}%
    \fi
  \fi
  \ifdim\descindentFB<\z@
    \ifFBListItemsAsPar
      \setlength{\descindentFB}{\labelindentFB}%
    \else
      \setlength{\descindentFB}{\listindentFB}%
    \fi
  \fi
}
```

\leftmarginFB
\FB@listHsettings

\FB@listHsettings holds the new horizontal settings chosen for French lists itemize, enumerate and description (two possible layouts).

```
\newdimen\leftmarginFB
\def\FB@listHsettings{%
  \ifFBListItemsAsPar
```

Optional layout: lists' items are typeset as paragraphs with indented labels.

```
    \itemindent=\labelindentFB
    \advance\itemindent by \labelwidthFB
    \advance\itemindent by \labelsep
    \leftmargini\z@
    \bbl@for\FB@dp {2, 3, 4, 5, 6}%
      {\csname leftmargin\romannumeral\FB@dp\endcsname =
         \labelindentFB}%
  \else
```

Default layout: labels hanging into the list left margin.

```
    \leftmarginFB=\labelwidthFB
    \advance\leftmarginFB by \labelsep
    \bbl@for\FB@dp {1, 2, 3, 4, 5, 6}%
      {\csname leftmargin\romannumeral\FB@dp\endcsname =
         \leftmarginFB}%
    \advance\leftmargini by \listindentFB
```

Same ‘parindent’ for paragraphs in lists’ items (was null as in standard lists).

```
    \listparindent=\parindent
  \fi
  \leftmargin=\csname leftmargin%
     \ifnum\@listdepth=\@ne i\else ii\fi\endcsname
}
```

\itemizeFB
\FB@itemizesettings

New environment for French itemize-lists. \FB@itemizesettings does two things: first suppress all vertical spaces including glue unless option StandardListSpacing is set, then set horizontal indentations according to \FB@listHsettings unless option ListOldLayout is true (compatibility with lists up to v2.5k).

```
\def\FB@itemizesettings{%
    \ifFBStandardListSpacing
    \else
      \FB@pardim=\parskip
      \ifdim\FB@pardim>\z@
        \setlength{\topsep}{-\FB@pardim}%
        \setlength{\partopsep}{\FB@pardim}%
        \setlength{\parsep}{\FB@pardim}%
        \setlength{\itemsep}{-\FB@pardim}%
      \else
        \setlength{\topsep}{\z@}%
        \setlength{\partopsep}{\z@}%
        \setlength{\parsep}{\z@}%
        \setlength{\itemsep}{\z@}%
      \fi
    \fi
    \settowidth{\labelwidth}{\csname\@itemitem\endcsname}%
    \ifFBListOldLayout
      \setlength{\leftmargin}{\labelwidth}%
      \addtolength{\leftmargin}{\labelsep}%
```

```
   \addtolength{\leftmargin}{\parindent}%
  \else
   \FB@listHsettings
  \fi
}
```

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

```
\def\itemizeFB{%
  \ifnum \@itemdepth >\thr@@\@toodeep\else
   \advance\@itemdepth by \@ne
   \edef\@itemitem{labelitem\romannumeral\the\@itemdepth}%
   \expandafter
   \listORI
   \csname\@itemitem\endcsname
   \FB@itemizesettings
  \fi
}
\let\enditemizeFB\endlistORI
```

The next function will be included in `\update@frenchlists` which is exectuted in `\extrasfrench{}` ‘AtBeginDocument’.

```
\def\setlabelitemsFB{%
 \let\labelitemi\Frlabelitemi
 \let\labelitemii\Frlabelitemii
 \let\labelitemiii\Frlabelitemiii
 \let\labelitemiv\Frlabelitemiv
 \ifdim\labelwidthFB<\z@
  \settowidth{\labelwidthFB}{\FrenchLabelItem}%
 \fi
}
```

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

```
\def\enumerateFB{%
 \ifnum \@enumdepth >\thr@@\@toodeep\else
  \advance\@enumdepth by \@ne
  \edef\@enumctr{enum\romannumeral\the\@enumdepth}%
  \expandafter
  \list
   \csname label\@enumctr\endcsname
   {\FB@listHsettings
    \usecounter\@enumctr\def\makelabel##1{\hss\llap{##1}}}%
 \fi
}
\let\endenumerateFB\endlistORI
```

\descriptionFB Same tuning for the description environment (see classes.dtx for the original definition). Customisable dimen \descindentFB, which defaults to \listindentFB, is added to \itemindent (first level only). When \descindentFB=0pt (1rst level labels start at the left margin), \leftmargini is reduced to \listindentFB instead of \listindentFB + \leftmarginFB.

When option ListItemsAsPar is turned to true, the description items are also displayed as paragraphs; \descindentFB=0pt can be used to push labels to the left margin.

```
\def\descriptionFB{%
      \list{}{\FB@listHsettings
              \labelwidth=\z@
              \ifFBListItemsAsPar
                \itemindent=\descindentFB
              \else
                \itemindent=-\leftmargin
                \ifnum\@listdepth=\@ne
                  \ifdim\descindentFB=\z@
                    \ifdim\listindentFB>\z@
                      \leftmargini=\listindentFB
                      \leftmargin=\leftmargini
                      \itemindent=-\leftmargin
                    \fi
                  \else
                    \advance\itemindent by \descindentFB
                  \fi
                \fi
              \fi
              \let\makelabel\descriptionlabel}%
}
\let\enddescriptionFB\endlistORI
```

\bbl@frenchlistlayout
\update@legacylists \update@legacylists will set up lists according to the final options (default or part of \frenchsetup{} eventually overruled in \FBprocess@options). This is for conventionnal lists *only*.

```
\def\update@legacylists{%
  \setlistindentFB
  \ifFBStandardListSpacing
  \else \let\list\listFB \fi
  \ifFBStandardItemizeEnv
  \else \let\itemize\itemizeFB \fi
  \ifFBStandardItemLabels
  \else \setlabelitemsFB \fi
  \ifFBStandardEnumerateEnv
  \else \let\enumerate\enumerateFB \let\description\descriptionFB \fi
}
```

Nothing has to be done at language's switches regarding lists, except at the first switch in case French is the main language, then lists are updated once for all. There is nothing to do for lists in \noextrasfrench.
Lists' layout no longer changes at language switches.

```
\def\bbl@frenchlistlayout{%
  \ifFB@mainlanguage@FR
    \ifFBnewlists
    \else
      \update@legacylists
      \let\update@legacylists\relax
    \fi
  \fi}
\addto\extrasfrench{\bbl@frenchlistlayout}
```

2.13 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.
Indentation changes at language switches only if IndentFirst=true and French isn't the main language.

```
\def\bbl@frenchindent{%
  \ifFBIndentFirst
    \ifFB@mainlanguage@FR\else\babel@save\@afterindentfalse\fi
    \let\@afterindentfalse\@afterindenttrue
    \@afterindenttrue
  \fi}
\addto\extrasfrench{\bbl@frenchindent}
```

2.14 Formatting footnotes

The layout of footnotes is controlled by two flags \ifFBAutoSpaceFootnotes and \ifFBFrenchFootnotes which are set by options of \frenchsetup{} (see section 2.11). 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!).
Common settings for both new and old code:

\parindentFFN
\dotFFN
\kernFFN

The value of \parindentFFN will be redefined at the \begin{document}, as the maximum of \parindent and 1.8em *unless* it has been set in the preamble (the weird value \maxdimen is for testing whether \parindentFFN has been set or not).

```
\newdimen\parindentFFN
\parindentFFN=\maxdimen
```

\FBfnindent will be set later on to the width of the box holding the footnote mark, \dotFFN and \kernFFN (flushed right). It is used by memoir and koma-script classes.

```
\newdimen\FBfnindent
\newcommand*{\dotFFN}{.}
\newcommand*{\kernFFN}{\kern .5em}
```

\FBfnmarkspace Let's define a customisable thin space which will be added before footnote's call.

```
\newcommand*{\FBfnmarkspace}{\kern .5\fontdimen2\font}
```

\newfootnotemarkFB This code is for the new footnotes templates.

```
\newcommand*{\newfootnotemarkFB}{%
  \setbox\@tempboxa\hbox{\@thefnmark}%
  \ifdim\wd\@tempboxa>\z@
    \llap{\@thefnmark}\dotFFN\kernFFN
  \fi}
```

This code is for legacy footnotes:

\@makefntextFB We 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), right aligned on \parindentFFN and followed by a dot and an half quad kern. 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).
\@makefntextFB's definition depends on the document's class.
Koma-script classes: they provide \deffootnote, a handy command to customise the footnotes' layout (see English manual scrguien.pdf); it redefines \@makefntext and \@@makefnmark. First, save the original definitions.

```
\ifFB@koma
  \let\@makefntextORI\@makefntext
  \let\@@makefnmarkORI\@@makefnmark
```

\@makefntextFB and \@@makefnmarkFB are used when option FrenchFootnotes is true.

```
  \deffootnote[\FBfnindent]{\z@}{\parindentFFN}%
              {\thefootnotemark\dotFFN\kernFFN}
  \let\@makefntextFB\@makefntext
  \let\@@makefnmarkFB\@@makefnmark
```

\@makefntextTH and \@@makefnmarkTH are meant for the \thanks command used by \maketitle when FrenchFootnotes is true.

```
  \deffootnote[\parindentFFN]{\z@}{\parindentFFN}%
              {\textsuperscript{\thefootnotemark}}
  \let\@makefntextTH\@makefntext
  \let\@@makefnmarkTH\@@makefnmark
```

Restore the original definitions.

```
  \let\@makefntext\@makefntextORI
  \let\@@makefnmark\@@makefnmarkORI
\fi
```

Definitions for the `memoir` class:

```
\@ifclassloaded{memoir}
```

(see original definition in `memman.pdf`)

```
   {\newcommand{\@makefntextFB}[1]{%
      \def\footscript##1{##1\dotFFN\kernFFN}%
      \setlength{\footmarkwidth}{\FBfnindent}%
      \setlength{\footmarksep}{-\footmarkwidth}%
      \setlength{\footparindent}{\parindentFFN}%
      \makefootmark #1}%
   }{}
```

Definitions for the `beamer` class:
the original definition is in `beamerbaseframecomponents.sty`, note that for the `beamer` class footnotes are LR-boxes, not paragraphs, so `\parindentFFN` is irrelevant.

```
\@ifclassloaded{beamer}
   {\def\@makefntextFB#1{%
      \def\insertfootnotetext{#1}%
      \def\insertfootnotemark{\insertfootnotemarkFB}%
      \usebeamertemplate***{footnote}}%
    \def\insertfootnotemarkFB{%
      \usebeamercolor[fg]{footnote mark}%
      \usebeamerfont*{footnote mark}%
      \llap{\@thefnmark}\dotFFN\kernFFN}%
   }{}
```

Now the default definition of `\@makefntextFB` for standard LaTeX and AMS classes. The next command prints the footnote mark according to the specifications of the French 'Imprimerie Nationale'. Keep in mind that `\@thefnmark` might be empty (i.e. in AMS classes' titles)!

```
\providecommand*{\insertfootnotemarkFB}{%
  \parindent=\parindentFFN
  \rule\z@\footnotesep
  \setbox\@tempboxa\hbox{\@thefnmark}%
  \ifdim\wd\@tempboxa>\z@
    \llap{\@thefnmark}\dotFFN\kernFFN
  \fi}
\providecommand\@makefntextFB[1]{\insertfootnotemarkFB #1}
```

The rest of `\@makefntext`'s customisation will be done at the `\begin{document}`: saving the original definition of `\@makefntext`, then redefining `\@makefntext` according to the value of flag `\ifFBFrenchFootnotes` (true or false).

\@footnotemark We will save the original definition of `\@footnotemark` at the `\begin{document}` in order to include any customisation that packages might have done; we define a variant `\@footnotemarkFB` which just adds a (customisable) thin space before the number or symbol calling a footnote (any space typed in is removed first). The choice between the two definitions (valid for the whole document) is controlled by flag `\ifFBAutoSpaceFootnotes`.
`\@footnotemark`'s customisation: `\FBfnmarkspace` will be added before footnote's call by `\@footnotemarkFB`.

```
\def\@footnotemarkFB{\leavevmode\unskip\unkern
                     \protect\FBfnmarkspace\@footnotemarkORI}%
```

The following command `\FBlegacyfootnote@switch` gathers the code needed to switch between French or Standard layout for footnotes; it is processed in `\FBprocess@options` just before `\begin{document}`.

```
\newcommand*{\FBlegacyfootnote@switch}{%
```

When the `footnotebackref` package is loaded, `babel-french` will not customise `\@footnotetext` in order to keep back referencing working.

```
  \IfPackageLoadedTF{footnotebackref}%
      {\FBFrenchFootnotesfalse
       \PackageWarning{french3.ldf}%
          {footnotebackref package loaded.\MessageBreak
           babel-french will NOT customise footnotes;%
           \MessageBreak reported}}%
      {}%
```

The `bigfoot` package deeply changes the way footnotes are handled. When `bigfoot` is loaded, we just warn the user that `babel-french` will not customise footnotes at all.

```
   \IfPackageLoadedTF{bigfoot}%
      {\PackageWarning{french.ldf}%
          {bigfoot package in use.\MessageBreak
           babel-french will NOT customise footnotes;%
           \MessageBreak reported}}%
```

Otherwise, footnotes may be customised according to the `\frenchsetup`{} options.

```
      {\let\@footnotemarkORI\@footnotemark
       \ifFBAutoSpaceFootnotes
         \let\@footnotemark\@footnotemarkFB
       \fi
       \ifdim\parindentFFN<\maxdimen
       \else
         \parindentFFN=\parindent
         \ifdim\parindentFFN<1.5em \parindentFFN=1.5em \fi
       \fi
       \settowidth{\FBfnindent}{\dotFFN\kernFFN}%
```

```
\addtolength{\FBfnindent}{\parindentFFN}%
\let\@makefntextORI\@makefntext
```

Koma-script classes require a special treatment.
Definition of **\@makefntext** for koma-script classes: running makefntextORI inside a group to reset **\localleftbox{}** and **\FBeverypar@quote** would mess up the layout of footnotes whenever the first manadatory argument of **\deffootnote{}** (used as **\leftskip**) is non-nil (default is 1em, 0pt in French).

```
\ifFB@koma
  \let\@@makefnmarkORI\@@makefnmark
  \long\def\@makefntext##1{%
    \let\FBeverypar@save\FBeverypar@quote
    \let\FBeverypar@quote\relax
    \ifFBFrenchFootnotes
      \ifx\footnote\thanks
        \let\@@makefnmark\@@makefnmarkTH
        \@makefntextTH{##1}
      \else
        \let\@@makefnmark\@@makefnmarkFB
        \@makefntextFB{##1}
      \fi
    \else
      \let\@@makefnmark\@@makefnmarkORI
      \@makefntextORI{##1}%
    \fi
    \let\FBeverypar@quote\FBeverypar@save
   }%
\else
```

Special add-on for the memoir class: **\@makefntext** is redefined as **\makethanksmark** by **\maketitle**, hence these settings to match the other notes' vertical alignment.

```
  \@ifclassloaded{memoir}%
     {\ifFBFrenchFootnotes
        \setlength{\thanksmarkwidth}{\parindentFFN}%
        \setlength{\thanksmarksep}{-\thanksmarkwidth}%
      \fi
     }{}%
```

Special add-on for the beamer class: issue a warning in case **\parindentFFN** has been changed.

```
  \@ifclassloaded{beamer}%
     {\ifFBFrenchFootnotes
        \ifdim\parindentFFN=1.5em\else
          \FBWarning{%
             \protect\parindentFFN\space is ineffective%
             \MessageBreak within the beamer class.%
             \MessageBreak Reported}%
        \fi
```

```
             \fi
            }{}%
```

Definition of `\@makefntext` for all other classes:

```
         \long\def\@makefntext##1{%
           \let\FBeverypar@save\FBeverypar@quote
           \let\FBeverypar@quote\relax
           \ifFBFrenchFootnotes
             \@makefntextFB{##1}%
           \else
             \@makefntextORI{##1}%
           \fi
           \let\FBeverypar@quote\FBeverypar@save
          }%
       \fi
    }%
  }
```

`\FBlegacyfootnote@switch` is executed when entering French for the first time (at `\begin{document}`), after possible redefinitions made by `latex-lab` for tagging.

```
\def\bbl@frenchfootnotes{%
  \ifFB@mainlanguage@FR
    \ifFBnewfootnotes
    \else
      \FBlegacyfootnote@switch
      \let\FBlegacyfootnote@switch\relax
    \fi
  \fi}
\addto\extrasfrench{\bbl@frenchfootnotes}
```

For compatibility reasons, we provide definitions for the commands dealing with the layout of footnotes in `babel-french` version 1.6. `\frenchsetup`{} (see in section 2.11) 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`.

```
\newcommand*{\AddThinSpaceBeforeFootnotes}{\FBAutoSpaceFootnotestrue}
\newcommand*{\FrenchFootnotes}{\FBFrenchFootnotestrue}
\newcommand*{\StandardFootnotes}{\FBFrenchFootnotesfalse}
```

2.15 Clean up and exit

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

```
\FBclean@on@exit
\ldf@finish\CurrentOption
\let\loadlocalcfg\FB@llc
```

```
</french>
```

2.16 Files `frenchb.ldf`, `francais.ldf`, `canadien.ldf` and `acadian.ldf`

Babel expects a <*lang*>`.ldf` file for each <*lang*>.
So we create portmanteau `.ldf` files for options `canadien`, `francais`, `frenchb` and `acadian`. These files themselves only load `french3.ldf` which does the real work. Warn users about options `acadien`, `canadien`, `frenchb` and `francais` being deprecated and force the recommended option `french`.

```
<*acadian>
\PackageInfo{acadian.ldf}%
  {Option 'acadian' for Babel is *deprecated*,\MessageBreak
   it might be removed sooner or later.  Please\MessageBreak
   use 'french' instead; reported}%
\chardef\l@acadian\l@french
\def\CurrentOption{french}
</acadian>
<*canadien>
\PackageInfo{canadien.ldf}%
  {Option 'canadien' for Babel is *deprecated*,\MessageBreak
   it might be removed sooner or later.  Please\MessageBreak
   use 'french' instead; reported}%
\chardef\l@canadien\l@french
\def\CurrentOption{french}
</canadien>
<*francais>
\PackageWarning{francais.ldf}%
  {Option 'francais' for Babel is *deprecated*,\MessageBreak
   it might be removed sooner or later.  Please\MessageBreak
   use 'french' instead; reported}%
\chardef\l@francais\l@french
\def\CurrentOption{french}
</francais>
<*frenchb>
\PackageWarning{frenchb}%
    {Option `frenchb' for Babel is *deprecated*,\MessageBreak
     it might be removed sooner or later.  Please\MessageBreak
     use `french' instead; reported}
\chardef\l@frenchb\l@french
\def\CurrentOption{french}
</frenchb>
```

3 Change History

Changes listed in reverse order (latest first) since v3.3 (2018).

v3.7b

General: \FBlegacyfootnote@switch moved to \extrasfrench (tagging issue). 69

New \ifFBnewlists and \ifFBnewfootnotes to handle the corresponding new templates. . . 44

\frenchsetup: New code to customise footnotes when the new templates are available. 54

v3.7a

General: Support for acadian dropped. The files acadian.ldf, canadien.ldf, frenchb.ldf and francais.ldf load french.ldf and print a warning. 70

\FB@xetex@punct@french: \XeTeXcharclass(es) of French double quotes are set in \FB@xetex@punct@french if options og=« and/or fg=» have been selected. 21

\frenchsetup: Option GlobalLayoutFrench deleted. . . . 47

\frquote: Flag \ifFBcloseguill does not apply to \@fgii. 28

v3.6c

\frenchsetup: Removed spurious @ in \FBCompactItemize@setup and \FBListOldLayout@setup commands' names. 47

v3.6b

\NoAutoSpacing: \NoAutoSpacing must be inhibited in bookmarks. . 26

v3.6a

General: Internal 'l3keys' replaces package 'keyval' for options' management. 44

\@footnotemark: Allow customisation of the space added in \@footnotemarkFB. 67

\degres: Simplify \degres definition for text and math mode: \textdegree always defined (TS1) since 2019. 35

v3.5s

General: Footnotes: no customising of \@footnotetext when the footnotebackref package is loaded. Just warn the user. 67

v3.5r

General: Compatibility with ucharclasses package added. . . 19

v3.5q

\listFB: Bug correction: \parsep should be related to \parskip and \listparindent to \parindent. . 58

v3.5p

\DecimalMathComma: \DecimalMathComma can again be used in the preamble for a global action. It now works as expected inside a group. 35

v3.5o

\FB@xetex@punct@french: \shorthandon and \shorthandoff are no longer redefined (it broke \shorthandoff*). 21

v3.5n

General: \FBGlobalLayoutFrench no longer set to false when French is not the main language. 45

\bbl@frenchindent: \bbl@frenchindent changed. \bbl@nonfrenchindent removed. 64

\bsc: Added command \bname (no small caps). 34

v3.5m

\FBtextellipsis: No longer redefine \dots, only \textellipsis's default definition is changed in French. 43

v3.5l

General: No warning about

\@makecaption for more classes. . 41
Redefine \fnum@figure and \fnum@table separately. 38

v3.5k

General: \degre, \degres, \circonflexe, \tild, \boi and \at are now safe in bookmarks. . 34
\pdfstringdefDisableCommands dropped. 58
Reorganise warnings about ':' in captions, according to enhancements in caption.sty v3.5a. 41
Small caps removed in \figurename and \tablename, use \fnum@figure and \fnum@table instead. 38
\bsc: \bsc now relies on \texorpdfstring to be safe in bookmarks. 34
\FB@fg: \FB@og and \FB@fg now rely on \texorpdfstring to be safe in bookmarks. 27
\frquote: \frquote now relies on \texorpdfstring to be safe in bookmarks. 29
\fup: \up and \fup now rely on \texorpdfstring to be safe in bookmarks. 31
\no: \no, \nos, \No, \Nos, \primo, \fprimo, now rely on \texorpdfstring to be safe in bookmarks. 33

v3.5j

General: For memoir, koma-script and beamer captions, \FB@std@sep has to be defined before activating the colon. 23

v3.5i

\frenchsetup: For memoir, koma-script and beamer classes, leave caption delimiter unchanged if it has been user customised. . . 56

v3.5e

General: StandardLayout and GlobalLayoutFrench options can no longer be toggled when French is not the main language. 45
\frquote: Make resettings global on exit. 30
new command \NoEveryParQuote. 30

v3.5d

\frenchsetup: ReduceListSpacing option depreciated: see StandardListSpacing. 47

v3.5c

General: Remove grouping inside \@makefntext, \localleftbox and \FBeverypar@quote saved and restored instead. 67
\frquote: \FBeverypar@quote's value now properly reset across level changes. 30
\noextrasfrench: \lccode of quote 0x27 changed from 0x2019 to 0x27 for Unicode engines. 17

v3.5b

General: Reset \FBeverypar@quote locally inside \@makefntext. Needed by \frquote. 67
\frquote: New command \FB@addquote@everypar to manage \everypar: \frquote failed when used immediately after a sectionning command. . . 29

v3.5a

General: New optional layout for lists: lists' items can be typeset as paragraphs with indented labels while the default leaves the labels hanging into the left margin. . . . 60
\descriptionFB: ListItemsAsPar option taken into account for description lists. 63
\frenchsetup: New option ListItemsAsPar for displaying lists' items "as paragraphs". 47

v3.4d

\frenchsetup: New test for deciding about utf8 encoding for keys og and fg (the former one fails with LaTeX 2018 release). 50

v3.4c

- \ifFBunicode: Reverting to former test, beware of \XeTeXrevision left as \relax by careless testing. 17

v3.4b

- \datefrench: Do not redefine \date as \frenchdate in French. 31

v3.4a

- General: \LdfInit checks \FBclean@on@exit instead of \captionsfrench (undefined in PLain). Prevents loading french.ldf again with acadian option. 15
  - babel-french now requires eTeX. . . 15
  - New command \FBsetspaces to fine tune spacing independently in French and in French dialects. . . 18
  - Patch for koma-script classes moved here, after \ifFBPartNameFull is defined, so that it applies to \extrasacadian too: \AtEndOfPackage is too late. 45
  - Shrink/stretch removed in \FBthousandsep. 38
- \datefrench: Specific code for Plain finally removed (babel bug reported). 31

v3.3c

- General: LaTeX 2017-04-15 defines TU encoding for Unicode engines, fontspec is no longer required. . . 58
  - New command \FBthousandsep to customise numprint. 38
  - New configurable kerns \FBmedkern, and \FBthickkern suitable for HTML translation. 33
  - Reorganise warnings when the caption, subcaption or floatrow packages are loaded before babel/french. 41
  - Reset \localleftbox locally inside \@makefntext. Needed by \frquote with LuaTeX. 67

v3.3b

- General: Generate portmanteau files acadian.ldf, canadien.ldf, frenchb.ldf, and francais.ldf and warn about deprecated options. 70
  - New 'if' \ifFBfrench to replace \iflanguage test which is based on patterns. 17

v3.3a

- General: Commands \frenchpartfirst, \frenchpartsecond and \frenchpartnameord added. 38
- \FBthinspace: Skips \FBcolonskip and \FBthinskip replaced by toks \FBcolonsp and \FBthinsp. 18
- \frenchsetup: \frenchbsetup is now an alias for \frenchsetup. 47
  - Options INGuillSpace, ThinColonSpace no longer delayed AtBeginDocument. 47
- \frquote: \FB@quotespace (kern), changed into \FB@guillspace. . . 29