

A Babel language definition file for French

frenchb3.dtx v3.7d, 2025-07-25

Daniel Flipo
daniel.flipo@free.fr

Contents

1	The French language	2
1.1	Basic interface	2
1.2	Customisation	5
1.2.1	<code>\frenchsetup</code>	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 <code>frenchb.ldf</code> , <code>français.ldf</code> , <code>canadien.ldf</code> and <code>acadian.ldf</code>	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 two 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`.

¹The file described in this section has version number v3.7d and was last revised on 2025-07-25.

²Always 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 language⁴, with the following effects:

1. French hyphenation patterns are made active;
2. ‘high punctuation’ characters (: ; ! ?) automatically add correct spacing⁵ 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

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

⁵Well, 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 1^{er} 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 3^{es}). 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°, 2°, 3°, 4°. `\FrenchEnumerate{6}` prints 6°.
6. Abbreviations for “Numéro(s)” and “numéro(s)” (N° N^{os} n° 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 `'1er 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 by 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 is 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:

<p>Text starting at ‘parindent’ \leq Leftmargin</p> <ul style="list-style-type: none"> – first item running on two lines or more... – first second level item on two lines... – next one... – second item... 	<p>Text starting at ‘parindent’ \leq Leftmargin</p> <ul style="list-style-type: none"> – first item running on two lines or more... – first second level item on two lines... – next one... – second item...
Default French layout	With <code>ListItemsAsPar=true</code>

`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*)`

⁶This 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` (defaults to a thin space) before ' ; ' ! ' ? ' or `\FBcolonspace` (defaults to `\space`) before ' : ' ; the defaults follow the French 'Imprimerie Nationale's recommendations. This is convenient in most cases but can lead to addition of spurious spaces in URLs, in MS-DOS paths or in timetables (10:55), except if they are typed in `\texttt` or verbatim mode. When the current font is a monospaced (typewriter) font, 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

⁷Unless 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 paragraph 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 paragraph 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~»`.

⁹Actually 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 up-casing 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 `français`.

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* precede 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); toggle 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 T_EX in your log-file; in French you should get with both 7-bit and 8-bit encodings
si-gnal contai-ner évé-ne-ment al-gèbre.

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

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

Frequent mismatches:

- you get sig-na1 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 (`13keys`). 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: redefinitions 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` (formerly `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 conversion 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 its own .ldf file; starting with version 3.3b, the main support for French is in french.ldf, portman-teau 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.

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

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

```
3 \def\fb@error#1#2{%
4   \begingroup
5     \newlinechar=`^^J
6     \def\{\^^J(french3.1df) }%
7     \errhelp{#2}\errmessage{\#1^^J}%
8   \endgroup}
9 \def\fb@warning#1{%
10  \begingroup
11    \newlinechar=`^^J
12    \def\{\^^J(french3.1df) }%
13    \message{\#1^^J}%
14  \endgroup}
15 \def\fb@info#1{%
16  \begingroup
17    \newlinechar=`^^J
18    \def\{\^^J}%
19    \wlog{#1}%
20  \endgroup}
```

Quit if eTeX is not available.

```
21 \let\bbl@tempa\relax
22 \begingroup\expandafter\expandafter\expandafter\endgroup
23 \expandafter\ifx\csname eTeXversion\endcsname\relax
24   \let\bbl@tempa\endinput
25   \fb@error{babel-french requires eTeX.\\
26             Aborting here}
27             {Original PlainTeX is not supported,\\
28             please use LuaTeX or XeTeX engines.}
29 \fi
30 \bbl@tempa
```

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

```
31 \let\bbl@tempa\relax
32 \ifdefined\babeltags
33 \else
34   \let\bbl@tempa\endinput
```

```

35 \ifdefined\PackageError
36   \PackageError{french.1df}
37     {babel-french requires babel v.3.16.\MessageBreak
38     Aborting here}
39     {Please upgrade Babel!}
40 \else
41   \fb@error{babel-french requires babel v.3.16.\
42     Aborting here}
43     {Please upgrade Babel!}
44 \fi
45 \fi
46 \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@<language>` also for eTeX and XeTeX formats which set `\lang@<language>`.

```

47 \def\FB@nopatterns{%
48   \ifdefined\l@nohyphenation
49     \addialect\l@french\l@nohyphenation
50     \edef\bbl@nulllanguage{\string\language=nohyphenation}%
51   \else
52     \edef\bbl@nulllanguage{\string\language=0}%
53     \addialect\l@french0
54   \fi
55   \@nopatterns{French}}
56 \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.

```

57 \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.

```

58 \newif\ifLaTeXe
59 \let\bbl@tempa\relax
60 \ifdefined\magnification
61 \else
62   \ifdefined\@compatibilitytrue
63     \LaTeXtrue
64   \else
65     \PackageError{french3.1df}
66       {LaTeX-2.09 format is no longer supported.\MessageBreak
67       Aborting here}
68       {Please upgrade to LaTeX2e!}
69     \let\bbl@tempa\endinput
70   \fi
71 \fi

```


72 `\bbl@tempa`

`\ifFBXeTeX` 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.

```
73 \newif\ifFBXeTeX
74 \newif\ifFBUnicode
75 \begingroup\expandafter\expandafter\expandafter\endgroup
76 \expandafter\ifx\csname XeTeXrevision\endcsname\relax
77 \else
78   \FBUnicode>true \FBXeTeX>true
79 \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=»}`.

```
80 \newif\ifFBfrench
```

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

In French, character “apostrophe” (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.

```
81 \def\extrasfrench{%
82   \FBfrench>true
83   \babel@savevariable{\lccode"27}%
84   \lccode"27="27
85   \ifFBUnicode
86     \babel@savevariable{\lccode"2019}%
87     \lccode"2019="2019
88   \fi
89 }
90 \def\noextrasfrench{\FBfrench>false}
```

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

```
91 \addto\extrasfrench{\bbl@frenchspacing}
92 \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.

```
93 \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.

```
94 \newcount\FB@stdchar
95 \newif\ifFB@xetex@punct
96 \ifdefined\XeTeXinterchartokenstate
97   \FB@xetex@puncttrue\FB@active@punctfalse
98   \ifdim\the\XeTeXversion\XeTeXrevision\p@ < 0.99994\p@
99     \chardef\FB@nonchar="FF \relax
100  \else
101    \chardef\FB@nonchar="FFF \relax
102  \fi
103  \FB@stdchar=\z@
104 \fi
```

`\FBguillspace` These three commands are meant for basic French. Other French dialects can use

`\FBcolonspace` different settings, see below. According to the I.N. specifications, the ‘:’ requires

`\FBthinspace` 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.

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

`\FBsetspaces` This command makes it easy to fine tune `\FBguillspace`, `\FBcolonspace` and `\FBthinspace` in French (default) or independently in a French dialect using the op-

tional argument. They are meant for LaTeX2e *only* and can only be used in the preamble. Four mandatory arguments¹⁰: 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".

```

110 \ifLaTeXe
111 \newcommand*\FBsetspaces}[5][french]{%
112   \namedef{FB#2space}{\hskip #3\fontdimen2\font
113     plus #4\fontdimen3\font
114     minus #5\fontdimen4\font \relax}}%
115 \@onlypreamble\FBsetspaces
116 \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.

```

117 \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 initial values will be restored when leaving French.

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

```

118 \newif\ifFB@og@fg@xetex

```

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

```

119 \ifFB@xetex@punct
120   \ifLaTeXe
121     \PackageInfo{french3.lfd}{No need for active punctuation
122                 characters\MessageBreak with this
123                 version of XeTeX!\MessageBreak reported}
124   \else
125     \fb@info{No need for active punctuation characters\\
126             with this version of XeTeX!}
127   \fi

```

Six new character classes are defined for babel-french.

```

128 \newXeTeXintercharclass\FB@punctthick
129 \newXeTeXintercharclass\FB@punctthin
130 \newXeTeXintercharclass\FB@punctnul
131 \newXeTeXintercharclass\FB@guilo
132 \newXeTeXintercharclass\FB@guilf
133 \newXeTeXintercharclass\FB@guilnul

```

As `\babel@savevariable` doesn't work inside a `\bb1@for` loop, we define a variant to save the `\XeTeXcharclass` values which will be modified in French.

```

134 \def\FBsavevariable@loop#1#2{\begingroup
135   \toks@\expandafter{\originalTeX #1}%
136   \edef\x{\endgroup
137     \def\noexpand\originalTeX{\the\toks@ #2=\the#1#2\relax}}%
138   \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.

```

139 \def\FB@charlist{"21,"3A,"3B,"3F,"AB,"BB,"28,"5B,"A0,"202F}
140 \def\FB@charlistUCC{}
141 \def\FB@charlistxeCJK{}
142 \edef\FB@charlistsave{\FB@charlist}
143 \ifLaTeXe
144   \AddToHook{env/document/before}{%
145     \IfPackageLoadedTF{ucharclasses}%
146     {\ifdefined\BasicLatinClass
147       \RenewCommandCopy{\FB@stdchar}{\BasicLatinClass}%
148       \def\FB@charlistUCC{"C0,"C2,"C6,"C7,"C8,"C9,"CA,"CB,"CE,"CF,%
149         "D4,"D6,"D9,"DB,"DC,"E0,"E2,"E6,"E7,"E8,"E9,"EA,"EB,"EE,%
150         "EF,"F4,"F6,"F9,"FB,"FC,"152,"153,"17F,"2019}%
151       \addto\FB@charlist{\FB@charlistUCC}%
152       \edef\FB@charlistsave{\FB@charlist}%
153     }
154   }%

```

```

155     \IfPackageLoadedTF{xeCJK}%
156     {\def\FB@charlistxeCJK{%
157         "29,"5D,"7B,"7D,"2C,"2D,"2E,"22,"25,"27,"60,"2019}%
158         \addto\FB@charlist{\FB@charlistxeCJK}%
159         \edef\FB@charlistsave{\FB@charlist}%
160     }}%
161 }
162 \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.

```

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

```

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

```

167     \ifx\FB@charlistUCC\@empty\else
168         \bbl@for\FB@char\FB@charlistUCC
169             {\XeTeXcharclass \FB@char \FB@stdchar}%
170     \fi

```

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

```

171     \ifx\FB@charlistxeCJK\@empty\else
172         \bbl@for\FB@char\FB@charlistxeCJK
173             {\XeTeXcharclass\FB@char=\FB@stdchar}%
174     \fi

```

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

```

175     \ifFB@og@fg@xetex
176         \XeTeXcharclass"13 = \FB@guilo
177         \XeTeXcharclass"14 = \FB@guilf
178         \XeTeXcharclass"AB = \FB@guilo
179         \XeTeXcharclass"BB = \FB@guilf
180         \XeTeXcharclass"A0 = \FB@guilnul
181         \XeTeXcharclass"202F = \FB@guilnul
182     \fi

```

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

```

183     \bbl@for\FB@char {\[,`[,`[,`["A0,"202F}%
184         {\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).

```

185 \XeTeXinterchartokenstate=\@ne
186 \XeTeXcharclass `\: = \FB@punctthick
187 \XeTeXinterchartoks \FB@stdchar \FB@punctthick = {%
188   \ifFB@spacing\ifhmode\FDP@colonspace\fi\fi}%
189 \XeTeXinterchartoks \FB@guilf \FB@punctthick = {%
190   \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.

```

191 \XeTeXinterchartoks \FB@nonchar \FB@punctthick = {%
192   \ifFB@spacing
193     \ifhmode
194       \ifdim\lastskip>1sp
195         \unskip\penalty\@M\FBcolonspace
196       \else
197         \FDP@colonspace
198       \fi
199   \fi
200 \fi}%
201 \bbl@for\FB@char {\`;\,`!\,`?}%
202   {\XeTeXcharclass\FB@char=\FB@punctthin}%
203 \XeTeXinterchartoks \FB@stdchar \FB@punctthin = {%
204   \ifFB@spacing\ifhmode\FDP@thinspace\fi\fi}%
205 \XeTeXinterchartoks \FB@guilf \FB@punctthin = {%
206   \ifFB@spacing\FDP@thinspace\fi}%
207 \XeTeXinterchartoks \FB@nonchar \FB@punctthin = {%
208   \ifFB@spacing
209     \ifhmode
210       \ifdim\lastskip>1sp
211         \unskip\penalty\@M\FBthinspace
212       \else
213         \FDP@thinspace
214       \fi
215     \fi
216   \fi}%
217 \ifFB@og@fg@xetex
218   \XeTeXinterchartoks \FB@guilo \FB@stdchar = {%
219     \ifFB@spacing\FB@guillspace\fi}%
220   \XeTeXinterchartoks \FB@guilo \FB@nonchar = {%
221     \ifFB@spacing\FB@guillspace\ignorespaces\fi}%
222   \XeTeXinterchartoks \FB@stdchar \FB@guilf = {%

```

```

223     \ifFB@spacing\FB@guillspace\fi}%
224     \XeTeXinterchartoks \FB@punctthin \FB@guilf = {%
225     \ifFB@spacing\FB@guillspace\fi}%
226     \XeTeXinterchartoks \FB@nonchar \FB@guilf = {%
227     \ifFB@spacing\unskip\FB@guillspace\fi}%
228     \fi
229   }
230   \addto\extrasfrench{\FB@xetex@punct@french}

```

End of specific code for punctuation with modern XeTeX engines.

```
231 \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 :

```

232 \newif\ifFB@koma
233 \ifLaTeXe
234   \@ifclassloaded{scrartcl}{\FB@komatrue}{}
235   \@ifclassloaded{scrbook}{\FB@komatrue}{}
236   \@ifclassloaded{scrreprt}{\FB@komatrue}{}
237   \ifFB@koma\def\FB@std@capsep{: \ } \fi
238   \@ifclassloaded{beamer}{\def\FB@std@capsep{: } }{}
239   \@ifclassloaded{memoir}{\def\FB@std@capsep{: }}{}
240 \fi

241 \ifFB@active@punct
242   \initiate@active@char{:}%
243   \initiate@active@char{;}%
244   \initiate@active@char{!}%
245   \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.

```

246   \declare@shorthand{french}{;};}%
247   \ifFB@spacing
248     \ifhmode
249       \ifdim\lastskip>1sp
250         \unskip\penalty\@M\FBthinspace
251       \else
252         \FDP@thinspace

```

```

253     \fi
254     \fi
255     \fi

```

Now we can insert a ; character.

```

256     \string;}

```

The next three definitions are very similar.

```

257 \declare@shorthand{french}{!}{}%
258   \ifFB@spacing
259     \ifhmode
260       \ifdim\lastskip>1sp
261         \unskip\penalty\@M\FBthinspace
262       \else
263         \FDP@thinspace
264       \fi
265     \fi
266   \fi
267   \string!}
268 \declare@shorthand{french}{?}{}%
269   \ifFB@spacing
270     \ifhmode
271       \ifdim\lastskip>1sp
272         \unskip\penalty\@M\FBthinspace
273       \else
274         \FDP@thinspace
275       \fi
276     \fi
277   \fi
278   \string?}
279 \declare@shorthand{french}{:}{}%
280   \ifFB@spacing
281     \ifhmode
282       \ifdim\lastskip>1sp
283         \unskip\penalty\@M\FBcolonspace
284       \else
285         \FDP@colonspace
286       \fi
287     \fi
288   \fi
289   \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.

```

290 \declare@shorthand{system}{:}{\string:}
291 \declare@shorthand{system}{!}{\string!}
292 \declare@shorthand{system}{?}{\string?}

```



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

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

```
294 \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.

```
295 \bbl@activate{:}\bbl@activate{;}%
296 \bbl@activate{!}\bbl@activate{?}%
297 }
298 \addto\noextrasfrench{%
299 \bbl@deactivate{:}\bbl@deactivate{;}%
300 \bbl@deactivate{!}\bbl@deactivate{?}%
301 }
302 \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’. its default value is true, but it can be set to false by `\frenchsetup{AutoSpacePunctuation=false}` for finer control.

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

`\AutoSpaceBeforeFDP` `\autospace@beforeFDP` and `\noautospace@beforeFDP` are internal commands. `\NoAutoSpaceBeforeFDP` `\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).

```
304 \def\autospace@beforeFDP{%
305 \def\FDP@thinspace{\penalty\@M\FBthinspace}%
306 \def\FDP@colonspace{\penalty\@M\FBcolonspace}}
307 \def\noautospace@beforeFDP{%
308 \let\FDP@thinspace\@empty
309 \let\FDP@colonspace\@empty}
310 \ifLaTeXe
311 \def\AutoSpaceBeforeFDP{\autospace@beforeFDP
312 \FBAutoSpacePunctuationtrue}
313 \def\NoAutoSpaceBeforeFDP{\noautospace@beforeFDP
314 \FBAutoSpacePunctuationfalse}
315 \AtEndOfPackage{\AutoSpaceBeforeFDP}
316 \else
317 \let\AutoSpaceBeforeFDP\autospace@beforeFDP
318 \let\NoAutoSpaceBeforeFDP\noautospace@beforeFDP
```

```

319 \AutoSpaceBeforeFDP
320 \fi

```

\rmfamilyFB In LaTeX2e **\ttfamily** (and hence **\texttt**) will be redefined ‘AtBeginDocument’ as **\sffamilyFB** **\ttfamilyFB** so that no space is added before the four ; : ! ? characters, even if **\ttfamilyFB** **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.

```

321 \ifLaTeXe
322 \DeclareRobustCommand\ttfamilyFB{\FB@spacingfalse \ttfamilyORI}
323 \DeclareRobustCommand\rmfamilyFB{\FB@spacingtrue \rmfamilyORI}
324 \DeclareRobustCommand\sffamilyFB{\FB@spacingtrue \sffamilyORI}
325 \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**.

```

326 \providecommand\texorpdfstring[2]{#1}
327 \DeclareRobustCommand*\NoAutoSpacing{}%
328 \texorpdfstring{\FB@spacingfalse
329 \ifFB@active@punct\shorthandoff{;:!?}\fi}}{}%
330 }

```

2.3 Commands for French quotation marks

\guillemotleft 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. **\guillemotright** In both cases the commands **\guillemotleft** and **\guillemotright** will print the **\textquoteddblleft** French opening and closing quote characters from the output font. For XeLaTeX **\textquoteddblright** **\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.

```

331 \ifLaTeXe
332 \else
333   \iffBunicode
334     \def\guillemotleft{\char"00AB}
335     \def\guillemotright{\char"00BB}
336     \def\textquotedblleft{\char"201C}
337     \def\textquotedblright{\char"201D}
338   \else
339     \def\guillemotleft{\leavevmode\raise0.25ex
340                       \hbox{\scriptscriptstyle\ll}}
341     \def\guillemotright{\raise0.25ex
342                       \hbox{\scriptscriptstyle\gg}}
343     \def\textquotedblleft{``}
344     \def\textquotedblright{' '}
345   \fi
346   \let\xspace\relax
347 \fi

```

\FB@og 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.

```

348 \newcommand*{\FB@og}{\texorpdfstring{\@FB@og}{\guillemotleft\space}}
349 \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.

```

350 \newcommand*{\FB@guillspace}{\penalty\@M\FBguillspace}
351 \iffB@xetex@punct
352   \DeclareRobustCommand*{\@FB@og}{\leavevmode
353     \bgroup\FB@spacingfalse\guillemotleft\egroup
354     \FB@guillspace}
355   \DeclareRobustCommand*{\@FB@fg}{\ifdim\lastskip>\z@\unskip\fi
356     \FB@guillspace
357     \bgroup\FB@spacingfalse\guillemotright\egroup}
358 \fi
359 \iffB@active@punct
360   \DeclareRobustCommand*{\@FB@og}{\leavevmode
361     \guillemotleft
362     \FB@guillspace}
363   \DeclareRobustCommand*{\@FB@fg}{\ifdim\lastskip>\z@\unskip\fi
364     \FB@guillspace
365     \guillemotright}
366 \fi

```

`\og` 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.

```
367 \newcommand*\og{\@empty}
368 \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).

```
369 \ifLaTeXe
370   \def\bbl@frenchguillemets{%
371     \renewcommand*\og{\FB@og}%
372     \renewcommand*\fg{\FB@fg\xspace}}
373   \renewcommand*\og{\textquotedblleft}
374   \renewcommand*\fg{\ifdim\lastskip>z@\unskip\fi
375     \textquotedblright\xspace}
376 \else
377   \def\bbl@frenchguillemets{\let\og\FB@og
378     \let\fg\FB@fg}
379   \def\og{\textquotedblleft}
380   \def\fg{\ifdim\lastskip>z@\unskip\fi\textquotedblright}
381 \fi

382 \addto\extrasfrench{\babel@save\og \babel@save\fg
383   \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...

```
384 \newcommand*\ogi{\FB@og}
385 \newcommand*\fgi{\FB@fg}
386 \newcommand*\@ogi{\ifmmode\hbox{\ogi}\else\ogi\fi}
387 \newcommand*\@fgi{\ifmmode\hbox{\fgi}\else\fgi\fi}
388 \newcommand*\ogii{\ifFBInnerGuillSingle
389   \guilsinglleft\FB@guillspace
390   \else \textquotedblleft
391   \fi}
392 \newcommand*\fgii{\ifFBInnerGuillSingle
393   \ifdim\lastskip>z@\unskip\fi
394   \FB@guillspace\guilsinglright
395   \else \textquotedblright}
```

```

396             \fi}
397 \newcommand*{\@ogii}{\ifmmode\hbox{\ogii}\else\ogii\fi}
398 \newcommand*{\@fgii}{\ifmmode\hbox{\fgii}\else\fgii\fi}

```

and the needed technical stuff to handle options:

```

399 \newcount\FBguill@level
400 \newtoks\FBold@everypar

```

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

```

401 \def\FB@addquote@everypar{%
402   \let\FBnew@everypar\everypar
403   \FBold@everypar=\expandafter{\the\everypar}%
404   \FBnew@everypar={\the\FBold@everypar\FBeverypar@quote}%
405   \let\everypar\FBold@everypar
406   \let\FB@addquote@everypar\relax
407 }
408 \newif\ifFBcloseguill \FBcloseguilltrue
409 \newif\ifFBInnerGuillSingle
410 \def\FBguillopen{\bgroup\NoAutoSpacing\guillemotleft\egroup}
411 \def\FBguillclose{\bgroup\NoAutoSpacing\guillemotright\egroup}
412 \let\FBguillnone\empty
413 \let\FBeveryparguill\FBguillopen
414 \let\FBverylineguill\FBguillnone
415 \let\FBeverypar@quote\relax
416 \let\FBveryline@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.

```

417 \ifLaTeXe
418   \DeclareRobustCommand\frquote{%
419     \texorpdfstring{\@ifstar{\FBcloseguillfalse\fr@quote}%
420                       {\FBcloseguilltrue \fr@quote}}%
421     {\bm@fr@quote}%
422   }
423   \newcommand{\bm@fr@quote}[1]{%
424     \guillemotleft\space #1\space\guillemotright}
425 \else
426   \newcommand\frquote[1]{\fr@quote{#1}}
427 \fi

```

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

```

428 \newcommand{\fr@quote}[1]{%
429   \leavevmode
430   \advance\FBguill@level by \@ne
431   \ifcase\FBguill@level
432     \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:

```

433   \ifx\FBeveryparguill\FBguillnone
434   \else
435     \def\FBeverypar@quote{\FBeveryparguill\FB@guillspace}%
436     \FB@addquote@everypar
437   \fi
438   \@ogi #1\@fgi
439 \or

```

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

```

440   \let\FBeverypar@quote\relax
441   \ifFBInnerGuillSingle
442     \ifx\FBeveryparguill\FBguilllopen
443       \def\FBeverypar@quote{\guillsinglleft\FB@guillspace}%
444     \fi
445   \fi
446   \ifx\FBeveryparguill\FBguillclose
447     \def\FBeverypar@quote{\guillsinglright\FB@guillspace}%
448   \fi
449   \@ogii #1\@fgii
450 \else

```

Warn if `\FBguill@level > 2`:

```

451   \ifx\PackageWarning\@undefined
452     \fb@warning{\noexpand\frquote\space handles up to
453               two levels.\\ Quotation not printed.}%
454   \else
455     \PackageWarning{french3.ldf}{%
456       \protect\frquote\space handles up to two levels.
457       \MessageBreak Quotation not printed. Reported}
458   \fi
459 \fi

```

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

```

460   \global\advance\FBguill@level by \m@ne
461   \ifcase\FBguill@level \global\let\FBeverypar@quote\relax
462   \or \gdef\FBeverypar@quote{\FBeveryparguill\FB@guillspace}%
463     \global\let\FBeveryline@quote\empty
464   \fi
465 }

```

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.

```

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

```

2.4 Date in French

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

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

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

```
478 \newcommand*{\FB@date}[3]{%
479   {{\number#3}\ifnum1=#3{\ier}\fi\FBdatespace
480   \cstype month\romannumeral#2name\endcstype
481   \ifx#1\@empty\else\FBdatespace\number#1\fi}}
482 \newcommand*{\FBdatebox}{\hbox}
483 \newcommand*{\FBdatespace}{\space}
484 \newcommand*{\frenchdate}{\FBdatebox\FB@date}
```

2.5 Extra utilities

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

`\up` `\up` eases the typesetting of superscripts like '1^{er}'. Up to version 2.0 of `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 `scalegnt` 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).

```
485 \newif\ifFB@poorman
486 \newdimen\FB@Mht
487 \ifLaTeXe
488   \AtEndOfPackage{\RequirePackage{scalegnt}}
```

`\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`.

```

489 \newcommand*\FBsupR}{-0.12}
490 \newcommand*\FBsupS}{0.65}
491 \newcommand*\FB@lc}[1]{\MakeLowercase{#1}}
492 \DeclareRobustCommand*\FB@up@fake}[1]{%
493   \settoheight{\FB@Mht}{M}%
494   \addtolength{\FB@Mht}{\FBsupR \FB@Mht}%
495   \addtolength{\FB@Mht}{-\FBsupS ex}%
496   \raisebox{\FB@Mht}{\scalefont{\FBsupS}{\FB@lc{#1}}}%
497 }

```

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.

```

498 \def\FB@split#1#2#3#4\@nil{\def\FB@firstthree{#1#2#3}%
499                               \def\FB@suffix{#4}}
500 \def\FB@x{x}
501 \def\FB@j{j}
502 \DeclareRobustCommand*\FB@up}[1]{%
503   \bgroup \FB@poormantrue
504   \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.

```

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

```



```

508     {\ifx\FB@suffix\FB@x \FB@poormanfalse\fi
509     \ifx\FB@suffix\FB@j \FB@poormanfalse\fi
510     \ifFB@poorman \FB@up@fake{#1}%
511     \else          \FB@up@real{#1}%
512     \fi}%
513     {\FB@up@fake{#1}}%
514   \egroup}

```

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

```

515   \newcommand*{\FB@up@real}[1]{\bgroup
516     \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.

```

517   \DeclareRobustCommand*{\fup}[1]{%
518     \texorpdfstring{\ifx\realsuperscript\@undefined
519       \FB@up{#1}%
520     \else
521       \bgroup\let\fakesuperscript\FB@up@fake
522       \realsuperscript{\FB@lc{#1}}\egroup
523     \fi
524     }{#1}%
525   }

```

Poor man's definition of `\up` for Plain.

```

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

```

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

```

\ier 529 \def\ieme{\up{e}\xspace}
\iere 530 \def\iemes{\up{es}\xspace}
\iemes 531 \def\ier{\up{er}\xspace}
\iers 532 \def\iers{\up{ers}\xspace}
\ieres 533 \def\iere{\up{re}\xspace}
534 \def\ieres{\up{res}\xspace}

```

`\FBmedkern`

```

\FBthickkern 535 \newcommand*{\FBmedkern}{\kern+.2em}
536 \newcommand*{\FBthickkern}{\kern+.3em}

```

`\primo` Some support macros relying on `\up` for numbering,

```

\primo) 537 \newcommand*{\FrenchEnumerate}[1]{%
\nos 538   #1\texorpdfstring{\up{o}\FBthickkern}{\textdegree\space}}
\Nos 539 \newcommand*{\FrenchPopularEnumerate}[1]{%
\No 540   #1\texorpdfstring{\up{o})\FBthickkern}{\textdegree\space}}
\no

```

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

```
541 \def\primo{\FrenchEnumerate1}
542 \def\secundo{\FrenchEnumerate2}
543 \def\tertio{\FrenchEnumerate3}
544 \def\quarto{\FrenchEnumerate4}
```

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

```
545 \def\fprimo{\FrenchPopularEnumerate1}
546 \def\fsecundo{\FrenchPopularEnumerate2}
547 \def\ftertio{\FrenchPopularEnumerate3}
548 \def\fquarto{\FrenchPopularEnumerate4}
```

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

```
549 \DeclareRobustCommand*\No}{%
550 \texorpdfstring{N\up{o}\FBmedkern}{N\textdegree\space}}
551 \DeclareRobustCommand*\no}{%
552 \texorpdfstring{n\up{o}\FBmedkern}{n\textdegree\space}}
553 \DeclareRobustCommand*\Nos}{%
554 \texorpdfstring{N\up{os}\FBmedkern}{N\textdegree\space}}
555 \DeclareRobustCommand*\nos}{%
556 \texorpdfstring{n\up{os}\FBmedkern}{n\textdegree\space}}
```

\bname These commands are meant to easily enter family names (in small capitals for the latter) while avoiding 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.

\bsc

```
557 \ifLaTeXe
558 \DeclareRobustCommand*\bname}[1]{%
559 \texorpdfstring{\leavevmode\begingroup\kern0pt #1\endgroup}{#1}}%
560 }
561 \DeclareRobustCommand*\bsc}[1]{%
562 \texorpdfstring{\leavevmode\begingroup\kern0pt \scshape #1\endgroup}%
563 {\textsc{#1}}}%
564 }
565 \else
566 \newcommand*\bname}[1]{\leavevmode\begingroup\kern0pt #1\endgroup}
567 \let\bsc\bname
568 \fi
```

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

```

569 \iffBunicode
570 \providecommand*\textbackslash{{\char"005C}}
571 \providecommand*\textasciicircum{{\char"005E}}
572 \providecommand*\textasciitilde{{\char"007E}}
573 \DeclareRobustCommand*\degre{{}^\circ}
574 \else
575 \DeclareRobustCommand*\degre{{\textdegree}}
576 \fi
577 \DeclareRobustCommand*\boi{{\textbackslash}}
578 \DeclareRobustCommand*\circonflexe{{\textasciicircum}}
579 \DeclareRobustCommand*\tild{{\textasciitilde}}
580 \newcommand*\at{{@}}

```

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

```

581 \DeclareRobustCommand*\degre{{\degre}}
582 \ifLaTeXe
583 \AtBeginDocument{%
584   \IfPackageLoadedTF{fontspec}{}{}%
585     {\DeclareRobustCommand*\degre{{}}%
586       \texorpdfstring{\hbox{\UseTextSymbol{TS1}{\textdegree}}}{%
587         {\textdegree}}}%
588   }%
589 }
590 \fi

```

2.6 Formatting numbers

\StandardMathComma As mentioned in the \TeX book p. 134, the comma is of type `\mathpunct` in math mode: **\DecimalMathComma** 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.

```

591 \newif\iffB@icomma
592 \newcount\mc@charclass
593 \newcount\mc@charfam
594 \newcount\mc@charslot
595 \newcount\std@mcc
596 \newcount\dec@mcc
597 \std@mcc=\mathcode`\,
598 \dec@mcc=\std@mcc
599 \@tempcnta=\std@mcc
600 \divide\@tempcnta by "1000

```

```

601 \multiply\@tempcnta by "1000
602 \advance\dec@mcc by -\@tempcnta
603 \newcommand*\dec@math@comma{\mathcode`\,\=\dec@mcc}
604 \newcommand*\std@math@comma{\mathcode`\,\=\std@mcc}
605 \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`.

```

606 \newif\if@FBpreamble
607 \ifLaTeXe \@FBpreambletrue \fi
608 \newif\if@preamble@DecimalMathComma
609 \newcommand*\DecimalMathComma{%
610   \if@FBpreamble \@preamble@DecimalMathCommatrue
611   \else
612     \ifFB@icomma
613       \PackageWarning{french3.1df}{%
614         icomma package loaded, \protect\DecimalMathComma\MessageBreak
615         does nothing. Reported}%
616     \else
617       \ifFBfrench
618         \dec@math@comma
619         \let\dec@m@c\dec@math@comma
620         \expandafter\addto\csname extras\language\endcsname
621         {\dec@m@c}%
622       \fi
623     \fi
624   \fi
625 }
626 \newcommand*\StandardMathComma{%
627   \ifFB@icomma
628     \PackageWarning{french3.1df}{%
629       icomma package loaded, \protect\StandardMathComma\MessageBreak
630       does nothing. Reported}%
631   \else
632     \ifFBfrench
633       \std@math@comma
634       \let\dec@m@c\relax
635     \fi
636   \fi
637 }

```

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

```

638 \ifLaTeXe\else
639   \addto\noextrasfrench{\std@math@comma}
640 \fi

```

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

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

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

```
643 \let\FBstop@here\relax
644 \def\FBclean@on@exit{%
645   \let\ifLaTeXe\iffalse
646   \let\LaTeXtrue\undefined
647   \let\LaTeXefalse\undefined
648   \let\FB@llc\loadlocalcfg
649   \let\loadlocalcfg\@gobble}
650 \ifx\magnification\@undefined
651 \else
652   \def\FBstop@here{%
653     \FBclean@on@exit
654     \ldf@finish\CurrentOption
655     \let\loadlocalcfg\FB@llc
656     \endinput}
657 \fi
658 \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`.

```
659 \AddToHook{env/document/before}{%
660   \@FBpreamblefalse
661   \IfPackageLoadedTF{icomma}%
662     {\FB@icommatrue
663       \if@preamble@DecimalMathComma
664         \PackageWarning{french3.lfd}{%
665           icomma package loaded, \protect\DecimalMathComma%
666           \MessageBreak does nothing. Reported}%
667       \fi
668     }%
669   {\if@preamble@DecimalMathComma
670     \ifFB@mainlanguage@FR \dec@math@comma \fi
671     \let\dec@m@c\dec@math@comma
672     \addto\extrasfrench{\dec@m@c}%
673   \fi}
```

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

```
674   \addto\noextrasfrench{\std@math@comma}%
675   }%
676 }
```

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.

```

677 \renewcommand*{\nombre}[1]{\Warning@nombre{#1}}
678 \newcommand*{\Warning@nombre}[1]{%
679   \ifdefined\numprint
680     \numprint{#1}%
681   \else
682     \PackageWarning{french3.ldf}{%
683       \protect\nombre\space now relies on package numprint.sty,%
684       \MessageBreak add \protect
685       \usepackage[autolanguage]{numprint},\MessageBreak
686       see file numprint.pdf for more options.\MessageBreak
687       \protect\nombre\space called}%
688     \global\let\Warning@nombre\relax
689     {#1}%
690   \fi
691 }

692 \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).

```

693 \StartBabelCommands*{french}{captions}
694   [unicode, fontenc=TU EU1 EU2, charset=utf8]
695   \SetString{\refname}{Références}
696   \SetString{\abstractname}{Résumé}
697   \SetString{\prefacename}{Préface}
698   \SetString{\contentsname}{Table des matières}
699   \SetString{\ccname}{Copie à }
700   \SetString{\proofname}{Démonstration}
701   \SetString{\partfirst}{Première}
702   \SetString{\partsecond}{Deuxième}
703   \SetStringLoop{ordinal#1}{%
704     \frenchpartfirst,\frenchpartsecond,Troisième,Quatrième,%
705     Cinquième,Sixième,Septième,Huitième,Neuvième,Dixième,Onzième,%
706     Douzième,Treizième,Quatorzième,Quinzième,Seizième,%
707     Dix-septième,Dix-huitième,Dix-neuvième,Vingtième}
708 \StartBabelCommands*{french}{captions}
709   \SetString{\refname}{R\ 'ef\ 'erences}

```

```

710 \SetString{\abstractname}{R'esum'e}
711 \SetString{\bibname}{Bibliographie}
712 \SetString{\prefacename}{Pr'eface}
713 \SetString{\chaptername}{Chapitre}
714 \SetString{\appendixname}{Annexe}
715 \SetString{\contentsname}{Table des mati`eres}
716 \SetString{\listfigurename}{Table des figures}
717 \SetString{\listtablename}{Liste des tableaux}
718 \SetString{\indexname}{Index}
719 \SetString{\figurename}{Figure}
720 \SetString{\tablename}{Table}
721 \SetString{\pagename}{page}
722 \SetString{\seename}{voir}
723 \SetString{\alsoname}{voir aussi}
724 \SetString{\enclname}{P.~J. }
725 \SetString{\ccname}{Copie `a }
726 \SetString{\headtoname}{}
727 \SetString{\proofname}{D'emonstration}
728 \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`.

```

729 \SetString{\partfirst}{Premi`ere}
730 \SetString{\partsecond}{Deuxi`eme}
731 \SetString{\partnameord}{partie}
732 \SetStringLoop{ordinal#1}{%
733   \partfirst,\partsecond,Troisi`eme,Quatri`eme,Cinqui`eme,%
734   Sixi`eme,Septi`eme,Huiti`eme,Neuvi`eme,Dixi`eme,%
735   Onzi`eme,Douzi`eme,Treizi`eme,Quatorzi`eme,Quinzi`eme,%
736   Seizi`eme,Dix-septi`eme,Dix-huiti`eme,Dix-neuvi`eme,%
737   Vingti`eme}
738 \AfterBabelCommands{%
739   \DeclareRobustCommand*{\FB@emptypart}{\def\thepart{\unskip}}%
740   \DeclareRobustCommand*{\FB@partname}{%
741     \iffBPartNameFull
742       \csname ordinal\romannumeral\value{part}\endcsname\space
743       \partnameord\FB@emptypart
744     \else
745       Partie%
746     \fi}%
747   }
748 \SetString{\partname}{\FB@partname}
749 \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`.

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

2.8 Figure and table captions

`\FBWarning` `\FBWarning` is an alias of `\PackageWarning{french3.1df}` which can be made silent by option `SuppressWarning`.

```
751 \newcommand{\FBWarning}[1]{\PackageWarning{french3.1df}{#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 precede 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.1df` makes '>' active).

```
752 \bgroup
753 \catcode`:=12 \catcode`>=12 \relax
754 \long\gdef\STD@makecaption#1#2{%
755   \vskip\abovecaptionskip
756   \sbox\@tempboxa{#1: #2}%
757   \ifdim \wd\@tempboxa >\hsize
758     #1: #2\par
759   \else
760     \global \@minipagefalse
761     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
762   \fi
763   \vskip\belowcaptionskip}
764 \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.

```
765 \newif\if@FBwarning@capsep
766 \iffB@active@punct\@FBwarning@capseptrue\fi
767 \newcommand*\CaptionSeparator{\space\textendash\space}
768 \def\FBCaption@Separator{: }
769 \long\def\FB@makecaption#1#2{%
770 \vskip\abovecaptionskip
771 \box\@tempboxa{#1\FBCaption@Separator #2}%
772 \ifdim \wd\@tempboxa >\hsize
773 #1\FBCaption@Separator #2\par
774 \else
775 \global \@minipagefalse
776 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
777 \fi
778 \vskip\belowcaptionskip}
```

Disable the standard warning with AMS and SMF classes.

```
779 \ifclassloaded{amsart}{\@FBwarning@capsepfalse}}
780 \ifclassloaded{amsbook}{\@FBwarning@capsepfalse}}
781 \ifclassloaded{amsdtx}{\@FBwarning@capsepfalse}}
782 \ifclassloaded{amslatex}{\@FBwarning@capsepfalse}}
783 \ifclassloaded{amproc}{\@FBwarning@capsepfalse}}
784 \ifclassloaded{smfart}{\@FBwarning@capsepfalse}}
785 \ifclassloaded{smfbook}{\@FBwarning@capsepfalse}}}
```

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

```
786 \ifclassloaded{IEEEconf}{\@FBwarning@capsepfalse}}
787 \ifclassloaded{IEEEtran}{\@FBwarning@capsepfalse}}
788 \ifclassloaded{revtex4-2}{\@FBwarning@capsepfalse}}
789 \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`)

```
790 \ifclassloaded{memoir}{\@FBwarning@capsepfalse}}
791 \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).

```
792 \ifclassloaded{beamer}{\@FBwarning@capsepfalse}}
793 \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’).

```
794 \AtBeginDocument{%
795   \ifx\@makecaption\STD@makecaption
796     \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`.

```
797   \ifFBOldFigTabCaptions
798   \else
799     \def\FBCaption@Separator{{\autospace@beforeFDP : }}%
800     \ifFBCustomiseFigTabCaptions
801       \ifFB@mainlanguage@FR
802         \def\FBCaption@Separator{\CaptionSeparator}%
803       \fi
804     \fi
805   \fi
806   \@FBwarning@capsepfalse
807 \fi
```

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

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

Final warning if relevant:

```
810   \if@FBwarning@capsep
811     \FBWarning
812     {Figures' and tables' captions might look like\MessageBreak
813     `Figure 1:' in French instead of `Figure 1 :'.\MessageBreak
814     If this happens, to fix this issue\MessageBreak
815     switch to LuaLaTeX or XeLaTeX or\MessageBreak
816     try to add \protect\usepackage{caption} or\MessageBreak
817     ... leave it as it is; reported}%
818   \fi
819   \let\FB@makecaption\relax
820   \let\STD@makecaption\relax
821 }
```

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.

```

822 \ifFBunicode
823 \else
824   \DeclareTextSymbol{\FBtextellipsis}{LY1}{133}
825   \DeclareTextCommand{\FBtextellipsis}{PU}{\9040\046}
826   \DeclareTextCommand{\FBtextellipsis}{PD1}{\203}
827   \DeclareTextCommandDefault{\FBtextellipsis}{%
828     .\kern\fontdimen3\font.\kern\fontdimen3\font.\xspace}%
829   \def\bbl@frenchdots{\babel@save\textellipsis
830     \let\textellipsis\FBtextellipsis}
831   \addto\extrasfrench{\bbl@frenchdots}
832 \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).

```

833 \ifFB@active@punct
834   \IfPackageLoadedTF{listings}
835     {\AddToHook{env/document/before}{%
836       \FBWarning{Please load the "listings" package\MessageBreak
837         AFTER babel/french; reported}}%
838     }{}
839 \fi

```

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

```

840 \newif\if@FBwarning@natbib
841 \ifFB@active@punct
842   \IfPackageLoadedTF{natbib}{}{\@FBwarning@natbibtrue}
843 \fi
844 \AddToHook{env/document/before}{%
845   \if@FBwarning@natbib
846     \IfPackageLoadedTF{natbib}{}{\@FBwarning@natbibfalse}%
847   \fi
848   \if@FBwarning@natbib
849     \FBWarning{Please load the "natbib" package\MessageBreak
850       BEFORE babel/french; reported}%
851   \fi
852 }

```

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

```

853 \newif\if@FBwarning@beamerarticle

```

```

854 \IfPackageLoadedTF{beamerarticle}{}{\@FBwarning@beamerarticletrue}
855 \AddToHook{env/document/before}{}%
856   \if@FBwarning@beamerarticle
857     \IfPackageLoadedTF{beamerarticle}{}%
858                                     {\@FBwarning@beamerarticlefalse}%
859   \fi
860   \if@FBwarning@beamerarticle
861     \FBWarning{Please load the "beamerarticle" package\MessageBreak
862               BEFORE babel/french; reported}%
863   \fi
864 }

```

2.11 Setup options: key/value stuff (l3keys)

Check LaTeX2e version (support for l3keys required).

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

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

```

866 \newif\ifFBnewlists
867 \newif\ifFBnewfootnotes
868 \IfPackageLoadedTF{latex-lab-testphase-block}{\FBnewliststrue}{}
869 \IfPackageLoadedTF{latex-lab-testphase-block}{\FBnewfootnotesttrue}{}

```

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 immediately 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).

```

870 \newif\ifFBShowOptions
871 \newif\ifFBStandardLayout           \FBStandardLayouttrue
872 \newif\ifFBStandardListSpacing     \FBStandardListSpacingtrue
873 \newif\ifFBListOldLayout
874 \newif\ifFBListItemsAsPar
875 \newif\ifFBCompactItemize
876 \newif\ifFBStandardItemizeEnv      \FBStandardItemizeEnvtrue
877 \newif\ifFBStandardEnumerateEnv    \FBStandardEnumerateEnvtrue
878 \newif\ifFBStandardItemLabels      \FBStandardItemLabelstrue
879 \newif\ifFBStandardLists           \FBStandardListstrue
880 \newif\ifFBIndentFirst

```

```

881 \newif\ifFBFrenchFootnotes
882 \newif\ifFBAutoSpaceFootnotes
883 \newif\ifFBOriginalTypewriter
884 \newif\ifFBThinColonSpace
885 \newif\ifFBThinSpaceInFrenchNumbers
886 \newif\ifFBFrenchSuperscripts      \FBFrenchSuperscriptstrue
887 \newif\ifFBLowercaseSuperscripts   \FBLowercaseSuperscriptstrue
888 \newif\ifFBPartNameFull             \FBPartNameFulltrue
889 \newif\ifFBCustomiseFigTabCaptions
890 \newif\ifFBOldFigTabCaptions
891 \newif\ifFBSmallCapsFigTabCaptions \FBSmallCapsFigTabCaptionstrue
892 \newif\ifBBSuppressWarning
893 \newif\ifBINGuillSpace

```

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

```

894 \ifFB@koma
895   \ifdefined\partformat
896     \def\FB@partformat@fix{%
897       \ifFBPartNameFull
898         \babel@save\partformat
899         \renewcommand*{\partformat}{\partname}%
900       \fi}
901   \addto\extrasfrench{\FB@partformat@fix}%
902 \fi
903 \fi

```

The default values of these flags are chosen so that `babel-french` does not change anything regarding the global layout. Some of them must be toggled 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.

```

904 \def\FB@french{french}
905 \newif\ifFB@mainlanguage@FR
906 \AtEndOfPackage{%
907   \ifx\bbl@main@language\FB@french \FB@mainlanguage@FRtrue \fi
908   \ifFB@mainlanguage@FR
909     \@ifclassloaded{beamer}%
910     {\PackageInfo{french3.1df}{%
911       No list customisation for the beamer class,%

```

```

912     \MessageBreak reported}}%
913     {\IfPackageLoadedTF{beamerarticle}%
914     {\FBStandardItemLabelsfalse
915     \FBStandardListSpacingfalse
916     \PackageInfo{french3.1df}}}%
917     Minimal list customisation for the beamerarticle%
918     \MessageBreak package; reported}}%

```

Otherwise customise lists “à la française”:

```

919     {\FBStandardListSpacingfalse
920     \FBStandardItemizeEnvfalse
921     \FBStandardEnumerateEnvfalse
922     \FBStandardItemLabelsfalse}%
923     }
924     \FBIndentFirsttrue
925     \FBFrenchFootnotestru
926     \FBAutoSpaceFootnotestru
927     \FBCustomiseFigTabCaptionstru
928     \fi
929 }

```

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

```

930 \DeclareKeys[FBsetup]
931 {
932   ShowOptions.if           = FBShowOptions           ,
933   StandardLayout.default:n = {true}           ,
934   StandardLayout.code      = \FBStandardLayout@setup{#1} ,
935   StandardListSpacing.if   = FBStandardListSpacing   ,
936   ReduceListSpacing.ifnot  = FBStandardListSpacing   ,
937   ListOldLayout.default:n  = {true}           ,
938   ListOldLayout.code       = \FBListOldLayout@setup{#1} ,
939   CompactItemize.default:n = {true}           ,
940   CompactItemize.code      = \FBCompactItemize@setup{#1} ,
941   StandardItemizeEnv.if    = FBStandardItemizeEnv   ,
942   StandardEnumerateEnv.if  = FBStandardEnumerateEnv ,
943   StandardItemLabels.if    = FBStandardItemLabels   ,
944   ItemLabels.store         = \FrenchLabelItem       ,
945   ItemLabeli.store         = \Frlabelitemi         ,
946   ItemLabelii.store        = \Frlabelitemii        ,
947   ItemLabeliii.store       = \Frlabelitemiii       ,
948   ItemLabeliv.store        = \Frlabelitemiv        ,
949   StandardLists.default:n  = {true}           ,
950   StandardLists.code       = \FBStandardLists@setup{#1} ,
951   ListItemsAsPar.if        = FBListItemsAsPar       ,
952   IndentFirst.if           = FBIndentFirst         ,
953   FrenchFootnotes.if       = FBFrenchFootnotes     ,
954   AutoSpaceFootnotes.if   = FBAutoSpaceFootnotes   ,

```

```

955 AutoSpacePunctuation.if = FBAutoSpacePunctuation ,
956 OriginalTypewriter.if = FBOriginalTypewriter ,
957 ThinColonSpace.default:n = {true} ,
958 ThinColonSpace.code = \FBThinColonSpace@setup{#1} ,
959 ThinSpaceInFrenchNumbers.if = FBThinSpaceInFrenchNumbers ,
960 FrenchSuperscripts.if = FBFrenchSuperscripts ,
961 LowercaseSuperscripts.if = FBLowercaseSuperscripts ,
962 PartNameFull.if = FBPartNameFull ,
963 CustomiseFigTabCaptions.if = FBCustomiseFigTabCaptions ,
964 OldFigTabCaptions.default:n = {true} ,
965 OldFigTabCaptions.code = \FBOldFigTabCaptions@setup{#1} ,
966 SmallCapsFigTabCaptions.default:n = {true} ,
967 SmallCapsFigTabCaptions.code = \FBSmallCapsFigTabCaptions@setup{#1} ,
968 SuppressWarning.default:n = {true} ,
969 SuppressWarning.code = \FBSuppressWarning@setup{#1} ,
970 INGuillSpace.default:n = {true} ,
971 INGuillSpace.code = \FBINGuillSpace@setup{#1} ,
972 InnerGuillSingle.if = FBInnerGuillSingle ,
973 EveryParGuill.default:n = {open} ,
974 EveryParGuill.code = \FBEveryParGuill@setup{#1} ,
975 og.code = \FBog@setup{#1} ,
976 fg.code = \FBfg@setup{#1} ,
977 }

```

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.

```

978 \newcommand*\frenchsetup}[1]{%
979   \SetKeys[FBsetup]{#1}%
980 }%
981 \@onlypreamble\frenchsetup

```

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

```

982 \let\frenchbsetup\frenchsetup
983 \@onlypreamble\frenchbsetup

```

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

```

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

```

```

994 \FBStandardListSpacingtrue
995 \FBStandardItemizeEnvtrue
996 \FBStandardItemLabelstrue
997 \FBStandardEnumerateEnvtrue
998 \FBIndentFirstfalse
999 \FBFrenchFootnotesfalse
1000 \FBAutoSpaceFootnotesfalse
1001 \else
1002 \FBStandardListSpacingfalse
1003 \FBStandardItemizeEnvfalse
1004 \FBStandardItemLabelsfalse
1005 \FBStandardEnumerateEnvfalse
1006 \FBIndentFirsttrue
1007 \FBFrenchFootnotesttrue
1008 \FBAutoSpaceFootnotesttrue
1009 \fi
1010 }
1011 \newcommand*\FBListOldLayout@setup}[1]%
1012 {\csname FBListOldLayout#1\endcsname
1013 \ifFBListOldLayout
1014 \FBStandardEnumerateEnvtrue
1015 \renewcommand*\FrenchLabelItem}{\textendash}%
1016 \fi
1017 }
1018 \newcommand*\FBCompactItemize@setup}[1]%
1019 {\csname FBCompactItemize#1\endcsname
1020 \ifFBCompactItemize
1021 \FBStandardItemizeEnvfalse
1022 \FBStandardEnumerateEnvfalse
1023 \else
1024 \FBStandardItemizeEnvtrue
1025 \FBStandardEnumerateEnvtrue
1026 \fi
1027 }
1028 \newcommand*\FBStandardLists@setup}[1]%
1029 {\csname FBStandardLists#1\endcsname
1030 \ifFBStandardLists
1031 \FBStandardListSpacingtrue
1032 \FBStandardItemizeEnvtrue
1033 \FBStandardEnumerateEnvtrue
1034 \FBStandardItemLabelstrue
1035 \else
1036 \FBStandardListSpacingfalse
1037 \FBStandardItemizeEnvfalse
1038 \FBStandardEnumerateEnvfalse
1039 \FBStandardItemLabelsfalse

```



```

1040 \fi
1041 }
1042 \newcommand*{\FBThinColonSpace@setup}[1]%
1043 {\csname FBThinColonSpace#1\endcsname
1044 \ifFBThinColonSpace
1045 \renewcommand*{\FBcolonspace}{\FBthinspace}%
1046 \fi
1047 }
1048 \newcommand*{\FBOldFigTabCaptions@setup}[1]%
1049 {\csname FBOldFigTabCaptions#1\endcsname
1050 \ifFBOldFigTabCaptions
1051 \def\FB@capsep@fix{\babel@save\FBCaption@Separator
1052 \def\FBCaption@Separator{\CaptionSeparator}}%
1053 \addto\extrasfrench{\FB@capsep@fix}%
1054 \fi
1055 }
1056 \newcommand*{\FBSmallCapsFigTabCaptions@setup}[1]%
1057 {\csname FBSmallCapsFigTabCaptions#1\endcsname
1058 \ifFBSmallCapsFigTabCaptions
1059 \else
1060 \let\FBfigtabshape\relax
1061 \fi
1062 }
1063 \newcommand*{\FBSuppressWarning@setup}[1]%
1064 {\csname FBSuppressWarning#1\endcsname
1065 \ifFBSuppressWarning
1066 \renewcommand{\FBWarning}[1]{}%
1067 \fi
1068 }
1069 \newcommand*{\FBINGuillSpace@setup}[1]%
1070 {\csname FBINGuillSpace#1\endcsname
1071 \ifFBINGuillSpace
1072 \renewcommand*{\FBguillspace}{\space}%
1073 \fi
1074 }
1075 \newcommand*{\FBEveryParGuill@setup}[1]%
1076 {\expandafter\let\expandafter
1077 \FBeveryparguill\csname FBguill#1\endcsname
1078 \ifx\FBeveryparguill\FBguillopen
1079 \else\ifx\FBeveryparguill\FBguillclose
1080 \else\ifx\FBeveryparguill\FBguillnone
1081 \else
1082 \let\FBeveryparguill\FBguillopen
1083 \FBWarning{Wrong value for `EveryParGuill':
1084 try `open',\MessageBreak
1085 `close' or `none'. Reported}%

```

```

1086         \fi
1087     \fi
1088 \fi
1089 }

```

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

```

1090 \def\FB@parse#1#2\endparse{\def\FB@second{#2}}%
1091 \newcommand*\FB@@og{%
1092   {\ifFBfrench
1093     \ifFB@spacing \FB@og\ignorespaces
1094     \else \guillemotleft
1095     \fi
1096   \else \guillemotleft
1097   \fi
1098   }
1099 \newcommand*\FB@@fg{%
1100   {\ifFBfrench
1101     \ifFB@spacing \FB@fg
1102     \else \guillemotright
1103     \fi
1104   \else \guillemotright
1105   \fi
1106   }
1107 \newcommand*\FBog@setup}[1]%
1108   {\ifFBunicode

```

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

```

1109     \FB@@og@fg@xetextrue
1110   \else

```

This is for conventional TeX engines:

```

1111     \AddToHook{env/document/before}{%
1112       \ifdefined\uc@dc1c

```

Package `inputenc` with `utf8x` (ucs) encoding loaded, use `\uc@dc1c`:

```

1113     \uc@dclc{171}{default}}{\FB@@og}%
1114     \else

```

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

```

1115     \FB@parse#1\endparse
1116     \ifx\FB@second\@empty

```

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

```

1117     \ifdefined\mule@def
1118     \mule@def{11}}{\FB@@og}%
1119     \else
1120     \ifdefined\DeclareInputText
1121     \@tempcnta`#1\relax
1122     \DeclareInputText{\the\@tempcnta}}{\FB@@og}%
1123     \else

```

Package inputenc not loaded, no way...

```

1124     \FBWarning{Option `og' requires package
1125     inputenc;\MessageBreak reported}%
1126     \fi
1127     \fi
1128     \else

```

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

```

1129     \DeclareUnicodeCharacter{00AB}}{\FB@@og}%
1130     \fi
1131     \fi}%
1132     \fi
1133     }

```

Same code for the closing quote.

```

1134 \newcommand*{\FBfg@setup}[1]%
1135 {\ifFBunicode
1136   \FB@og@fg@xetextrue
1137   \else
1138   \AddToHook{env/document/before}}{%
1139   \ifdefined\uc@dclc
1140   \uc@dclc{187}{default}}{\FB@@fg}%
1141   \else
1142   \FB@parse#1\endparse
1143   \ifx\FB@second\@empty
1144   \ifdefined\mule@def
1145   \mule@def{27}}{\FB@@fg}}%
1146   \else
1147   \ifdefined\DeclareInputText
1148   \@tempcnta`#1\relax
1149   \DeclareInputText{\the\@tempcnta}}{\FB@@fg}%
1150   \else

```

```

1151         \FBWarning{Option `fg' requires package
1152             inputenc;\MessageBreak reported}%
1153         \fi
1154     \fi
1155     \else
1156         \DeclareUnicodeCharacter{00BB}{\FB@@fg}%
1157     \fi
1158 \fi}%
1159 \fi
1160 }
1161 % \end{macro}
1162 %
1163 % \begin{macro}{\FBprocess@options}
1164 % |\FBprocess@options| will be executed at |\begin{document}|:
1165 % it first checks about packages loaded in the preamble (possibly
1166 % after \babel) which customise lists: currently \pkg{enumitem},
1167 % \pkg{paralist} and \pkg{enumerate}; then it processes the options
1168 % as set by \fbsetup{} or forced for compatibility with packages
1169 % loaded in the preamble.
1170 %
1171 % When French is the main language, |\extrasfrench| and
1172 % |\captionsfrench| are executed by \babel{} at |\begin{document}|,
1173 % i.e.\ after |\FBprocess@options|.
1174 % \begin{macrocode}
1175 \newcommand*{\FBprocess@options}{%
Update flags if a package customising lists has been loaded, currently: enumitem,
paralist, enumerate.
1176 \IfPackageLoadedTF{enumitem}{%
1177     \ifFBStandardItemizeEnv
1178     \else
1179         \FBStandardItemizeEnvtrue
1180         \PackageInfo{french3.ldf}%
1181         {Setting StandardItemizeEnv=true for\MessageBreak
1182             compatibility with enumitem package,\MessageBreak
1183             reported}%
1184     \fi
1185     \ifFBStandardEnumerateEnv
1186     \else
1187         \FBStandardEnumerateEnvtrue
1188         \PackageInfo{french3.ldf}%
1189         {Setting StandardEnumerateEnv=true for\MessageBreak
1190             compatibility with enumitem package,\MessageBreak
1191             reported}%
1192     \fi}}%
1193 \IfPackageLoadedTF{paralist}{%
1194     \ifFBStandardItemizeEnv

```

```

1195 \else
1196   \FBStandardItemizeEnvtrue
1197   \PackageInfo{french3.1df}%
1198   {Setting StandardItemizeEnv=true for\MessageBreak
1199   compatibility with paralist package,\MessageBreak
1200   reported}%
1201 \fi
1202 \ifFBStandardEnumerateEnv
1203 \else
1204   \FBStandardEnumerateEnvtrue
1205   \PackageInfo{french3.1df}%
1206   {Setting StandardEnumerateEnv=true for\MessageBreak
1207   compatibility with paralist package,\MessageBreak
1208   reported}%
1209 \fi}}%
1210 \IfPackageLoadedTF{enumerate}}%
1211 \ifFBStandardEnumerateEnv
1212 \else
1213   \FBStandardEnumerateEnvtrue
1214   \PackageInfo{french3.1df}%
1215   {Setting StandardEnumerateEnv=true for\MessageBreak
1216   compatibility with enumerate package,\MessageBreak
1217   reported}%
1218 \fi}}%
1219 \ifFB@mainlanguage@FR
1220 \else
1221   \ifFBStandardItemizeEnv
1222   \else
1223     \PackageWarning{french3.1df}%
1224     {babel-french will not customise lists' layout\MessageBreak
1225     when French is not the main language,\MessageBreak
1226     reported}%
1227   \fi
1228 \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{}`).

```

1229 \ifFBnewlists
1230   \FBWarning{You requested LaTeX tagging support.\MessageBreak
1231   Babel-french's list customization is currently\MessageBreak
1232   incompatible with the new lists' implementation\MessageBreak
1233   (still experimental) required to support tagging.%
1234   \MessageBreak Babel-french's list customization is *DISABLED*%
1235   \MessageBreak when tagging is enabled (see frenchb.pdf).%
1236   \MessageBreak Reported
1237   }%

```

```

1238 \setlistindentFB
1239 \fi

```

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

```

1240 \ifFBnewfootnotes
1241 \ifdim\parindentFFN<\maxdimen
1242 \else
1243 \parindentFFN=\parindent
1244 \ifdim\parindentFFN<1.8em \parindentFFN=1.8em \fi
1245 \fi
1246 \settowidth{\FBfnindent}{\dotFFN\kernFFN}%
1247 \addtolength{\FBfnindent}{\parindentFFN}%
1248 \ifFBFrenchFootnotes
1249 \NewSocketPlug{fntext/mark}{french}
1250 {\hb@xt@ \footnotemargin{\hss\newfootnotemarkFB}}
1251 \AssignSocketPlug{fntext/mark}{french}
1252 \AddToHook{cmd/maketitle/before}
1253 {\AssignSocketPlug{fntext/mark}{default}}
1254 \AddToHook{cmd/maketitle/after}
1255 {\AssignSocketPlug{fntext/mark}{french}}
1256 \AddToHook{env/minipage/begin}
1257 {\AssignSocketPlug{fntext/mark}{default}}
1258 \AddToHook{fntext/para}{\parindent=\parindentFFN}
1259 \AddToHook{fntext/para}{\let\FBeverypar@quote\relax}
1260 \fi
1261 \ifFBAutoSpaceFootnotes
1262 \AddToHook{fnmark/before}{\FBfnmarkspace}
1263 \fi
1264 \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 commands, we use `\fnum@figure` and `\fnum@table` when available (not in `beamer.cls` f.i.).

```

1265 \ifx\FBfigtabshape\relax
1266 \else
1267 \ifdefined\fnum@figure
1268 \let\fnum@figureORI\fnum@figure
1269 \renewcommand{\fnum@figure}{\ifFBfrench\FBfigtabshape\fi
1270 \fnum@figureORI}}%
1271 \fi
1272 \ifdefined\fnum@table
1273 \let\fnum@tableORI\fnum@table
1274 \renewcommand{\fnum@table}{\ifFBfrench\FBfigtabshape\fi
1275 \fnum@tableORI}}%

```

```
1276 \fi
1277 \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.

```
1278 \iffBAutoSpacePunctuation
1279 \autospace@beforeFDP
1280 \else
1281 \noautospace@beforeFDP
1282 \fi
```

When **OriginalTypewriter** is set to **false** (the default), `\ttfamily`, `\rmfamily` and `\sffamily` are redefined as `\ttfamilyFB`, `\rmfamilyFB` and `\sffamilyFB` respectively to prevent addition of automatic spaces before the four active characters in computer code.

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

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.

```
1292 \IfPackageLoadedTF{numprint}%
1293 {\ifnprt@autolanguage
1294 \providecommand*\npstylefrench{}%
1295 \iffBThinSpaceInFrenchNumbers
1296 \renewcommand*\FBthousandsep{\FBthinspace}%
1297 \fi
1298 \g@addto@macro\npstylefrench{\npthousandsep{\FBthousandsep}}%
1299 \fi
1300 }{}%
```

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.

```
1301 \iffBFrenchSuperscripts
1302 \DeclareRobustCommand*\up{}%
1303 \texorpdfstring{\@ifstar{\FB@up@fake}{\fup}}{}%
1304 }
1305 \else
```

```

1306   \DeclareRobustCommand*\up}{%
1307     \texorpdfstring{\@ifstar{\FB@up@fake}{\textsuperscript}}}%
1308   }
1309 \fi

```

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

```

1310 \iffBLowercaseSuperscripts
1311 \else
1312   \renewcommand*\FB@lc}[1]{##1}%
1313 \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`.

```

1314 \iffB@koma
1315   \ifx\captionformat\FB@std@capsep
1316     \iffBCustomiseFigTabCaptions
1317       \renewcommand*\captionformat{\CaptionSeparator}%
1318     \else
1319       \renewcommand*\captionformat{\autospace@beforeFDP : }%
1320     \fi
1321 \fi
1322 \fi
1323 \@ifclassloaded{memoir}%
1324   {\ifx\@contdelim\FB@std@capsep
1325     \iffBCustomiseFigTabCaptions
1326       \captiondelim{\CaptionSeparator}%
1327     \else
1328       \captiondelim{\autospace@beforeFDP : }%
1329     \fi
1330   \fi}%
1331 \@ifclassloaded{beamer}%
1332   {\protected@edef\FB@capsep{%
1333     \csname beamer@@@tpl@caption label separator\endcsname}%
1334   \ifx\FB@capsep\FB@std@capsep
1335     \iffBCustomiseFigTabCaptions
1336       \defbeamertemplate{caption label separator}{FBcustom}%
1337       \CaptionSeparator}%
1338     \setbeamertemplate{caption label separator}[FBcustom]%
1339   \else
1340     \defbeamertemplate{caption label separator}{FBcolon}%
1341     {\autospace@beforeFDP : }%
1342     \setbeamertemplate{caption label separator}[FBcolon]%
1343   \fi
1344 \fi}%

```


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

```
1345 \iffBShowOptions
1346 \GenericWarning{* }{%
1347 *** List of possible options for babel-french ***\MessageBreak
1348 [Default values between brackets when french is loaded *LAST*]%
1349 \MessageBreak
1350 ShowOptions [false]\MessageBreak
1351 StandardLayout [false]\MessageBreak
1352 PartNameFull [true]\MessageBreak
1353 IndentFirst [true]\MessageBreak
1354 ListItemsAsPar [false]\MessageBreak
1355 StandardListSpacing [false]\MessageBreak
1356 StandardItemizeEnv [false]\MessageBreak
1357 StandardEnumerateEnv [false]\MessageBreak
1358 StandardItemLabels [false]\MessageBreak
1359 ItemLabels=\textendash, \textbullet,
1360 \protect\ding{43},... [\textendash]\MessageBreak
1361 ItemLabeli=\textendash, \textbullet,
1362 \protect\ding{43},... [\textendash]\MessageBreak
1363 ItemLabelii=\textendash, \textbullet,
1364 \protect\ding{43},... [\textendash]\MessageBreak
1365 ItemLabeliii=\textendash, \textbullet,
1366 \protect\ding{43},... [\textendash]\MessageBreak
1367 ItemLabeliv=\textendash, \textbullet,
1368 \protect\ding{43},... [\textendash]\MessageBreak
1369 StandardLists [false]\MessageBreak
1370 ListOldLayout [false]\MessageBreak
1371 FrenchFootnotes [true]\MessageBreak
1372 AutoSpaceFootnotes [true]\MessageBreak
1373 AutoSpacePunctuation [true]\MessageBreak
1374 ThinColonSpace [false]\MessageBreak
1375 OriginalTypewriter [false]\MessageBreak
1376 og= <left quote character>, fg= <right quote character>%
1377 INGuillSpace [false]\MessageBreak
1378 EveryParGuill=open, close, none [open]\MessageBreak
1379 InnerGuillSingle [false]\MessageBreak
1380 ThinSpaceInFrenchNumbers [false]\MessageBreak
1381 SmallCapsFigTabCaptions [true]\MessageBreak
1382 CustomiseFigTabCaptions [true]\MessageBreak
1383 OldFigTabCaptions [false]\MessageBreak
1384 FrenchSuperscripts [true]\MessageBreak
1385 LowercaseSuperscripts [true]\MessageBreak
1386 SuppressWarning [false]\MessageBreak
1387 \MessageBreak
1388 *****%
1389 \MessageBreak\protect\frenchsetup{ShowOptions}}
```

```

1390 \fi
1391 }

```

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. Force `\FBprocess@options` to be processed only once.

```

1392 \AddToHook{env/document/before}{%
1393   \providecommand*\xspace{\relax}%
1394   \FBprocess@options
1395   \renewcommand*\FBprocess@options{\relax}%

```

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*`!

```

1396 \begingroup
1397   \newcommand{\FBOTone}{OT1}%
1398   \ifx\encodingdefault\FBOTone
1399     \FBWarning{OT1 encoding should not be used for French.%
1400               \MessageBreak
1401               Add \protect\usepackage[T1]{fontenc} to the
1402               preamble\MessageBreak of your document; reported}%
1403   \fi
1404 \endgroup
1405 }

```

2.12 French lists

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

```

1406 \let\listORI\list
1407 \let\endlistORI\endlist
1408 \newdimen\FB@pardim
1409 \def\FB@listVsettings{%
1410   \setlength{\topsep}{0.8ex plus 0.4ex minus 0.4ex}%

```

```

1411 \setlength{\partopsep}{0.4ex plus 0.2ex minus 0.2ex}%
1412 \setlength{\itemsep}{0.4ex plus 0.2ex minus 0.2ex}%
1413 \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`.

```
1414 \FB@pardim=\parskip
```

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

```

1415 \ifdim\FB@pardim>\z@
1416 \addtolength{\topsep}{-\FB@pardim}%
1417 \addtolength{\partopsep}{\FB@pardim}%
1418 \setlength{\parsep}{\FB@pardim}%
1419 \addtolength{\itemsep}{-\FB@pardim}%
1420 \fi
1421 }
1422 \def\listFB#1#2{\listORI{#1}{\FB@listVsettings #2}}
1423 \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` Default labels for French itemize-lists —same label for all levels—, (already defined as `\Frlabelitemi` empty by `\DeclareKey{}`):

```

\Frlabelitemii 1424 \renewcommand*{\FrenchLabelItem}{\textemdash}
\Frlabelitemiii 1425 \renewcommand*{\Frlabelitemi}{\FrenchLabelItem}
\Frlabelitemiv 1426 \renewcommand*{\Frlabelitemii}{\FrenchLabelItem}
1427 \renewcommand*{\Frlabelitemiii}{\FrenchLabelItem}
1428 \renewcommand*{\Frlabelitemiv}{\FrenchLabelItem}

```

`\listindentFB` Let’s define four dimens `\listindentFB`, `\descindentFB`, `\labelindentFB` and `\descindentFB` `\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 reasonable defaults later when entering French (see below)

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

```
1429 \newdimen\listindentFB
1430 \setlength{\listindentFB}{-1pt}
1431 \newdimen\descindentFB
1432 \setlength{\descindentFB}{-1pt}
1433 \newdimen\labelindentFB
1434 \setlength{\labelindentFB}{-1pt}
1435 \newdimen\labelwidthFB
1436 \setlength{\labelwidthFB}{-1pt}
```

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

```
1437 \def\setlistindentFB{%
1438   \ifdim\labelindentFB<\z@
1439     \ifdim\parindent=\z@
1440       \setlength{\labelindentFB}{1.5em}%
1441     \else
1442       \setlength{\labelindentFB}{\parindent}%
1443     \fi
1444   \fi
1445   \ifdim\listindentFB<\z@
1446     \ifdim\parindent=\z@
1447       \setlength{\listindentFB}{1.5em}%
1448     \else
1449       \setlength{\listindentFB}{\parindent}%
1450     \fi
1451   \fi
1452   \ifdim\descindentFB<\z@
1453     \ifFBListItemsAsPar
1454       \setlength{\descindentFB}{\labelindentFB}%
1455     \else
1456       \setlength{\descindentFB}{\listindentFB}%
1457     \fi
1458   \fi
1459 }
```

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

```
1460 \newdimen\leftmarginFB
1461 \def\FB@listHsettings{%
1462   \ifFBListItemsAsPar
```

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

```
1463   \itemindent=\labelindentFB
1464   \advance\itemindent by \labelwidthFB
```

```

1465 \advance\itemindent by \labelsep
1466 \leftmargini\z@
1467 \bb1@for\FB@dp {2, 3, 4, 5, 6}%
1468 {\csname leftmargin\romannumeral\FB@dp\endcsname =
1469 \labelindentFB}%
1470 \else

```

Default layout: labels hanging into the list left margin.

```

1471 \leftmarginFB=\labelwidthFB
1472 \advance\leftmarginFB by \labelsep
1473 \bb1@for\FB@dp {1, 2, 3, 4, 5, 6}%
1474 {\csname leftmargin\romannumeral\FB@dp\endcsname =
1475 \leftmarginFB}%
1476 \advance\leftmargini by \listindentFB

```

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

```

1477 \listparindent=\parindent
1478 \fi
1479 \leftmargin=\csname leftmargin%
1480 \ifnum\@listdepth=\@ne i\else ii\fi\endcsname
1481 }

```

\itemizeFB New environment for French itemize-lists.

\FB@itemizesettings **\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).

```

1482 \def\FB@itemizesettings{%
1483 \ifFBStandardListSpacing
1484 \else
1485 \FB@pardim=\parskip
1486 \ifdim\FB@pardim>\z@
1487 \setlength{\topsep}{-\FB@pardim}%
1488 \setlength{\partopsep}{\FB@pardim}%
1489 \setlength{\parsep}{\FB@pardim}%
1490 \setlength{\itemsep}{-\FB@pardim}%
1491 \else
1492 \setlength{\topsep}{\z@}%
1493 \setlength{\partopsep}{\z@}%
1494 \setlength{\parsep}{\z@}%
1495 \setlength{\itemsep}{\z@}%
1496 \fi
1497 \fi
1498 \settowidth{\labelwidth}{\csname\@itemitem\endcsname}%
1499 \ifFBListOldLayout
1500 \setlength{\leftmargin}{\labelwidth}%
1501 \addtolength{\leftmargin}{\labelsep}%

```

```

1502     \addtolength{\leftmargin}{\parindent}%
1503     \else
1504     \FB@listHsettings
1505     \fi
1506 }

```

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

```

1507 \def\itemizeFB{%
1508     \ifnum \@itemdepth >\thr@@\@toodeep\else
1509     \advance\@itemdepth by \@ne
1510     \edef\@itemitem{labelitem\romannumeral\the\@itemdepth}%
1511     \expandafter
1512     \listORI
1513     \csname\@itemitem\endcsname
1514     \FB@itemizesettings
1515     \fi
1516 }
1517 \let\enditemizeFB\endlistORI

```

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

```

1518 \def\setlabelitemsFB{%
1519     \let\labelitemi\Frlabelitemi
1520     \let\labelitemii\Frlabelitemii
1521     \let\labelitemiii\Frlabelitemiii
1522     \let\labelitemiv\Frlabelitemiv
1523     \ifdim\labelwidthFB<\z@
1524     \settowidth{\labelwidthFB}{\FrenchLabelItem}%
1525     \fi
1526 }

```

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

```

1527 \def\enumerateFB{%
1528     \ifnum \@enumdepth >\thr@@\@toodeep\else
1529     \advance\@enumdepth by \@ne
1530     \edef\@enumctr{enum\romannumeral\the\@enumdepth}%
1531     \expandafter
1532     \list
1533     \csname label\@enumctr\endcsname
1534     {\FB@listHsettings
1535     \usecounter\@enumctr\def\makelabel##1{\hss\llap{##1}}}%
1536     \fi

```

```

1537 }
1538 \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` (1st 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.

```

1539 \def\descriptionFB{%
1540     \list{}{\FB@listHsettings
1541         \labelwidth=\z@
1542         \ifFBListItemsAsPar
1543             \itemindent=\descindentFB
1544         \else
1545             \itemindent=-\leftmargin
1546             \ifnum\@listdepth=\@ne
1547                 \ifdim\descindentFB=\z@
1548                     \ifdim\listindentFB>\z@
1549                         \leftmargini=\listindentFB
1550                         \leftmargin=\leftmargini
1551                         \itemindent=-\leftmargin
1552                     \fi
1553                 \else
1554                     \advance\itemindent by \descindentFB
1555                 \fi
1556             \fi
1557         \let\makelabel\descriptionlabel}%
1558 }
1559 }
1560 \let\enddescriptionFB\endlistORI

```

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

```

1561 \def\update@legacylists{%
1562     \setlistindentFB
1563     \ifFBStandardListSpacing
1564     \else \let\list\listFB \fi
1565     \ifFBStandardItemizeEnv
1566     \else \let\itemize\itemizeFB \fi
1567     \ifFBStandardItemLabels
1568     \else \setlabelitemsFB \fi
1569     \ifFBStandardEnumerateEnv

```

```

1570 \else \let\enumerate\enumerateFB \let\description\descriptionFB \fi
1571 }

```

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.

```

1572 \def\bb1@frenchlistlayout{%
1573   \iffB@mainlanguage@FR
1574     \iffBnewlists
1575     \else
1576       \update@legacylists
1577       \let\update@legacylists\relax
1578     \fi
1579   \fi}
1580 \addto\extrasfrench{\bb1@frenchlistlayout}

```

2.13 French indentation of sections

`\bb1@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.

```

1581 \def\bb1@frenchindent{%
1582   \iffBIndentFirst
1583     \iffB@mainlanguage@FR\else\babel@save\@afterindentfalse\fi
1584     \let\@afterindentfalse\@afterindenttrue
1585     \@afterindenttrue
1586   \fi}
1587 \addto\extrasfrench{\bb1@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` 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 `\dotFFN` `\kernFFN` `\maxdimen` is for testing whether `\parindentFFN` has been set or not).

```

1588 \newdimen\parindentFFN
1589 \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.

```
1590 \newdimen\FBfnindent
1591 \newcommand*\dotFFN}{.}
1592 \newcommand*\kernFFN}{\kern .5em}
```

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

```
1593 \newcommand*\FBfnmarkspace}{\kern .5\fontdimen2\font}
```

`\newfootnotemarkFB` This code is for the new footnotes templates.

```
1594 \newcommand*\newfootnotemarkFB}{%
1595   \setbox\@tempboxa\hbox{\@thefnmark}%
1596   \ifdim\wd\@tempboxa>\z@
1597     \llap{\@thefnmark}\dotFFN\kernFFN
1598   \fi}
```

This code is for legacy footnotes:

`\@makefnmarkFB` We define `\@makefnmarkFB`, a variant of `\@makefnmark` 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).

`\@makefnmarkFB`'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 `\@makefnmark` and `\@@makefnmark`. First, save the original definitions.

```
1599 \ifFB@koma
1600   \let\@makefnmarkORI\@makefnmark
1601   \let\@@makefnmarkORI\@@makefnmark
```

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

```
1602 \deffootnote[\FBfnindent]{\z@}{\parindentFFN}%
1603   {\thefnmark\dotFFN\kernFFN}
1604 \let\@makefnmarkFB\@makefnmark
1605 \let\@@makefnmarkFB\@@makefnmark
```

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

```
1606 \deffootnote[\parindentFFN]{\z@}{\parindentFFN}%
1607   {\textsuperscript{\thefnmark}}
1608 \let\@makefnmarkTH\@makefnmark
1609 \let\@@makefnmarkTH\@@makefnmark
```

Restore the original definitions.

```
1610 \let\@makefntext\@makefntextORI
1611 \let\@@makefnmark\@@makefnmarkORI
1612 \fi
```

Definitions for the memoir class:

```
1613 \ifclassloaded{memoir}
```

(see original definition in memman.pdf)

```
1614 {\newcommand{\@makefntextFB}[1]{%
1615   \def\footscript###1{###1\dotFFN\kernFFN}%
1616   \setlength{\footmarkwidth}{\FBfnindent}%
1617   \setlength{\footmarksep}{-\footmarkwidth}%
1618   \setlength{\footparindent}{\parindentFFN}%
1619   \makefootmark #1}%
1620 }
```

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.

```
1621 \ifclassloaded{beamer}
1622 {\def\@makefntextFB#1{%
1623   \def\insertfootnotetext{#1}%
1624   \def\insertfootnotemark{\insertfootnotemarkFB}%
1625   \usebeamertemplate***{footnote}}%
1626 \def\insertfootnotemarkFB{%
1627   \usebeamercolor[fg]{footnote mark}%
1628   \usebeamerfont*{footnote mark}%
1629   \llap{\@thefnmark}\dotFFN\kernFFN}%
1630 }
```

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)!

```
1631 \providecommand*\insertfootnotemarkFB{%
1632   \parindent=\parindentFFN
1633   \rule\z@\footnotesep
1634   \setbox\@tempboxa\hbox{\@thefnmark}%
1635   \ifdim\wd\@tempboxa>\z@
1636     \llap{\@thefnmark}\dotFFN\kernFFN
1637   \fi}
1638 \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`.

```
1639 \def\@footnotemarkFB{\leavevmode\unskip\unkern
1640           \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}`.

```
1641 \newcommand*\FBlegacyfootnote@switch{}%
```

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

```
1642 \IfPackageLoadedTF{footnotebackref}%
1643   {\FBFrenchFootnotesfalse
1644     \PackageWarning{french3.1df}%
1645       {footnotebackref package loaded.\MessageBreak
1646         babel-french will NOT customise footnotes;%
1647         \MessageBreak reported}}%
1648   {}%
```

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.

```
1649 \IfPackageLoadedTF{bigfoot}%
1650   {\PackageWarning{french.1df}%
1651     {bigfoot package in use.\MessageBreak
1652       babel-french will NOT customise footnotes;%
1653       \MessageBreak reported}}%
```

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

```
1654   {\let\@footnotemarkORI\@footnotemark
1655     \ifFBAutoSpaceFootnotes
1656       \let\@footnotemark\@footnotemarkFB
1657     \fi
1658     \ifdim\parindentFFN<\maxdimen
1659     \else
1660       \parindentFFN=\parindent
1661       \ifdim\parindentFFN<1.5em \parindentFFN=1.5em \fi
1662     \fi
1663     \settowidth{\FBfnindent}{\dotFFN\kernFFN}%
```

```

1664     \addtolength{\FBfnindent}{\parindentFFN}%
1665     \let\@makefntextORI\@makefntext

```

Koma-script classes require a special treatment.

Definition of `\@makefntext` for koma-script classes:

```

1666     \iffB@koma
1667     \let\@@makefnmarkORI\@@makefnmark
1668     \long\def\@makefntext###1{%
1669         \let\FBeverypar@save\FBeverypar@quote
1670         \let\FBeverypar@quote\relax
1671         \iffBFrenchFootnotes
1672             \ifx\footnote\thanks
1673                 \let\@@makefnmark\@@makefnmarkTH
1674                 \@makefntextTH{###1}
1675             \else
1676                 \let\@@makefnmark\@@makefnmarkFB
1677                 \@makefntextFB{###1}
1678             \fi
1679         \else
1680             \let\@@makefnmark\@@makefnmarkORI
1681             \@makefntextORI{###1}%
1682         \fi
1683         \let\FBeverypar@quote\FBeverypar@save
1684     }%
1685 \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.

```

1686     \@ifclassloaded{memoir}%
1687     {
1688         \iffBFrenchFootnotes
1689         \setlength{\thanksmarkwidth}{\parindentFFN}%
1690         \setlength{\thanksmarksep}{-\thanksmarkwidth}%
1691     }%

```

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

```

1692     \@ifclassloaded{beamer}%
1693     {
1694         \iffBFrenchFootnotes
1695         \ifdim\parindentFFN=1.5em\else
1696             \FBWarning{%
1697                 \protect\parindentFFN\space is ineffective%
1698                 \MessageBreak within the beamer class.%
1699                 \MessageBreak Reported}%
1700         \fi
1701     }%

```

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

```
1702     \long\def\@makefntext###1{%
1703         \let\FBeverypar@save\FBeverypar@quote
1704         \let\FBeverypar@quote\relax
1705         \ifFBFrenchFootnotes
1706             \@makefntextFB{###1}%
1707         \else
1708             \@makefntextORI{###1}%
1709         \fi
1710         \let\FBeverypar@quote\FBeverypar@save
1711     }%
1712 \fi
1713 }%
1714 }
```

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

```
1715 \def\bb1@frenchfootnotes{%
1716     \ifFB@mainlanguage@FR
1717         \ifFBnewfootnotes
1718         \else
1719             \FBlegacyfootnote@switch
1720             \let\FBlegacyfootnote@switch\relax
1721         \fi
1722     \fi}
1723 \addto\extrasfrench{\bb1@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`.

```
1724 \newcommand*\AddThinSpaceBeforeFootnotes{\FBAutoSpaceFootnotestrue}
1725 \newcommand*\FrenchFootnotes{\FBFrenchFootnotestrue}
1726 \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.

```
1727 \FBclean@on@exit
1728 \ldf@finish\CurrentOption
1729 \let\loadlocalcfg\FB@llc
1730 </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 `acadian`, `canadien`, `frenchb` and `francais` being deprecated and force the recommended option `french`.

```
1731 <*acadian>
1732 \PackageInfo{acadian.ldf}%
1733 {Option 'acadian' for Babel is *deprecated*,\MessageBreak
1734   it might be removed sooner or later. Please\MessageBreak
1735   use 'french' instead; reported}%
1736 \chardef\l@acadian\l@french
1737 \def\CurrentOption{french}
1738 </acadian>
1739 <*canadien>
1740 \PackageInfo{canadien.ldf}%
1741 {Option 'canadien' for Babel is *deprecated*,\MessageBreak
1742   it might be removed sooner or later. Please\MessageBreak
1743   use 'french' instead; reported}%
1744 \chardef\l@canadien\l@french
1745 \def\CurrentOption{french}
1746 </canadien>
1747 <*francais>
1748 \PackageWarning{francais.ldf}%
1749 {Option 'francais' for Babel is *deprecated*,\MessageBreak
1750   it might be removed sooner or later. Please\MessageBreak
1751   use 'french' instead; reported}%
1752 \chardef\l@francais\l@french
1753 \def\CurrentOption{french}
1754 </francais>
1755 <*frenchb>
1756 \PackageWarning{frenchb}%
1757 {Option `frenchb' for Babel is *deprecated*,\MessageBreak
1758   it might be removed sooner or later. Please\MessageBreak
1759   use `french' instead; reported}
1760 \chardef\l@frenchb\l@french
1761 \def\CurrentOption{french}
1762 </frenchb>
```

3 Change History

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

v3.7d	General: <code>\FBprocess@options</code> must be processed only once.	58	of the space added in <code>\@footnotemarkFB</code>	67	
v3.7b	General: <code>\FBlegacyfootnote@switch</code> moved to <code>\extrasfrench</code> (tagging issue).	69	<code>\degres</code> : Simplify <code>\degres</code> definition for text and math mode: <code>\textdegree</code> always defined (TS1) since 2019.	35	
	New <code>\ifFBnewlists</code> and <code>\ifFBnewfootnotes</code> to handle the corresponding new templates. . .	44	v3.5s	General: Footnotes: no customising of <code>\@footnotetext</code> when the <code>footnotebackref</code> package is loaded. Just warn the user.	67
	<code>\frenchsetup</code> : New code to customise footnotes when the new templates are available.	54	v3.5r	General: Compatibility with <code>ucharclasses</code> package added. . .	19
v3.7a	General: Support for acadian dropped. The files <code>acadian.ldf</code> , <code>canadien.ldf</code> , <code>frenchb.ldf</code> and <code>francais.ldf</code> load <code>french.ldf</code> and print a warning.	70	v3.5q	<code>\listFB</code> : Bug correction: <code>\parsep</code> should be related to <code>\parskip</code> and <code>\listparindent</code> to <code>\parindent</code> . . .	58
	<code>\FB@xetex@punct@french</code> : <code>\XeTeXcharclass(es)</code> of French double quotes are set in <code>\FB@xetex@punct@french</code> if options <code>og=«</code> and/or <code>fg=»</code> have been selected.	21	v3.5p	<code>\DecimalMathComma</code> : <code>\DecimalMathComma</code> can again be used in the preamble for a global action. It now works as expected inside a group.	35
	<code>\frenchsetup</code> : Option <code>GlobalLayoutFrench</code> deleted. . .	47	v3.5o	<code>\FB@xetex@punct@french</code> : <code>\shorthandon</code> and <code>\shorthandoff</code> are no longer redefined (it broke <code>\shorthandoff*</code>).	21
	<code>\frquote</code> : Flag <code>\ifFBcloseguill</code> does not apply to <code>\@fgii</code>	28	v3.5m	General: <code>\FBGlobalLayoutFrench</code> no longer set to false when French is not the main language.	45
v3.6c	<code>\frenchsetup</code> : Removed spurious <code>@</code> in <code>\FBCompactItemize@setup</code> and <code>\FBListOldLayout@setup</code> commands' names.	47	<code>\bbl@frenchindent</code> : <code>\bbl@frenchindent</code> changed. <code>\bbl@nonfrenchindent</code> removed.	64	
v3.6b	<code>\NoAutoSpacing</code> : <code>\NoAutoSpacing</code> must be inhibited in bookmarks. .	26	<code>\bsc</code> : Added command <code>\bname</code> (no small caps).	34	
v3.6a	General: Internal 'l3keys' replaces package 'keyval' for options' management.	44	<code>\FBtextellipsis</code> : No longer redefine <code>\dots</code> , only <code>\textellipsis</code> 's default definition is changed in		
	<code>\@footnotemark</code> : Allow customisation				

French.	43	v3.5e	General: StandardLayout and GlobalLayoutFrench options can no longer be toggled when French is not the main language.	45
v3.5l			<code>\frquote</code> : Make resettings global on exit.	30
General: No warning about <code>\@makecaption</code> for more classes.	41		new command <code>\NoEveryParQuote</code>	30
Redefine <code>\fnum@figure</code> and <code>\fnum@table</code> separately.	38	v3.5d	<code>\frenchsetup</code> : ReduceListSpacing option deprecated: see StandardListSpacing.	47
v3.5k		v3.5c	General: Remove grouping inside <code>\@makefntext</code> , <code>\localleftbox</code> and <code>\FBeverypar@quote</code> saved and restored instead.	67
General: <code>\degre</code> , <code>\degres</code> , <code>\circonflexe</code> , <code>\tild</code> , <code>\boi</code> and <code>\at</code> are now safe in bookmarks.	34		<code>\frquote</code> : <code>\FBeverypar@quote</code> 's value now properly reset across level changes.	30
<code>\pdfstringdefDisableCommands</code> dropped.	58		<code>\noextrasfrench</code> : <code>\lcode</code> of quote 0x27 changed from 0x2019 to 0x27 for Unicode engines.	17
Reorganise warnings about ‘:’ in captions, according to enhancements in <code>caption.sty</code> v3.5a.	41	v3.5b	General: Reset <code>\FBeverypar@quote</code> locally inside <code>\@makefntext</code> . Needed by <code>\frquote</code>	67
Small caps removed in <code>\figurename</code> and <code>\tablename</code> , use <code>\fnum@figure</code> and <code>\fnum@table</code> instead.	38		<code>\frquote</code> : New command <code>\FB@addquote@everypar</code> to manage <code>\everypar</code> : <code>\frquote</code> failed when used immediately after a sectioning command.	29
<code>\bsc</code> : <code>\bsc</code> now relies on <code>\texorpdfstring</code> to be safe in bookmarks.	34	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
<code>\FB@fg</code> : <code>\FB@og</code> and <code>\FB@fg</code> now rely on <code>\texorpdfstring</code> to be safe in bookmarks.	27		<code>\descriptionFB</code> : ListItemsAsPar option taken into account for description lists.	63
<code>\frquote</code> : <code>\frquote</code> now relies on <code>\texorpdfstring</code> to be safe in bookmarks.	29		<code>\frenchsetup</code> : New option ListItemsAsPar for displaying lists’ items “as paragraphs”.	47
<code>\fup</code> : <code>\up</code> and <code>\fup</code> now rely on <code>\texorpdfstring</code> to be safe in bookmarks.	31	v3.4d	<code>\frenchsetup</code> : New test for deciding	
<code>\no</code> : <code>\no</code> , <code>\nos</code> , <code>\No</code> , <code>\Nos</code> , <code>\primo</code> , <code>\fprimo</code> , now rely on <code>\texorpdfstring</code> to be safe in bookmarks.	33			
v3.5j				
General: For memoir, koma-script and beamer captions, <code>\FB@std@sep</code> has to be defined before activating the colon.	23			
v3.5i				
<code>\frenchsetup</code> : For memoir, koma-script and beamer classes, leave caption delimiter unchanged if it has been user customised.	56			

about utf8 encoding for keys og and fg (the former one fails with LaTeX 2018 release).	50	New configurable kerns <code>\FBmedkern</code> , and <code>\FBthickkern</code> suitable for HTML translation.	33
v3.4c		Reorganise warnings when the caption, subcaption or floatrow packages are loaded before babel/french.	41
<code>\ifFBunicode</code> : Reverting to former test, beware of <code>\XeTeXrevision</code> left as <code>\relax</code> by careless testing.	17	Reset <code>\localleftbox</code> locally inside <code>\@makefn</code> text. Needed by <code>\frquote</code> with LuaTeX.	67
v3.4b		v3.3b	
<code>\datefrench</code> : Do not redefine <code>\date</code> as <code>\frenchdate</code> in French.	31	General: Generate portmanteau files <code>acadian.ldf</code> , <code>canadien.ldf</code> , <code>frenchb.ldf</code> , and <code>francais.ldf</code> and warn about deprecated options.	70
v3.4a		New 'if' <code>\ifFBfrench</code> to replace <code>\iflanguage</code> test which is based on patterns.	17
General: <code>\LdfInit</code> checks		v3.3a	
<code>\FBclean@on@exit</code> instead of <code>\captionfrench</code> (undefined in PLain). Prevents loading <code>french.ldf</code> again with <code>acadian</code> option.	15	General: Commands <code>\frenchpartfirst</code> , <code>\frenchpartsecond</code> and <code>\frenchpartnameord</code> added.	38
babel-french now requires eTeX.	15	<code>\FBthinspace</code> : Skips <code>\FBcolonskip</code> and <code>\FBthinskip</code> replaced by <code>\FBcolonsp</code> and <code>\FBthinsp</code>	18
New command <code>\FBsetspaces</code> to fine tune spacing independently in French and in French dialects.	18	<code>\frenchsetup</code> : <code>\frenchbsetup</code> is now an alias for <code>\frenchsetup</code>	47
Patch for koma-script classes moved here, after <code>\ifFBPartNameFull</code> is defined, so that it applies to <code>\extrasacadian</code> too: <code>\AtEndOfPackage</code> is too late.	45	Options <code>INGuillSpace</code> , <code>ThinColonSpace</code> no longer delayed <code>AtBeginDocument</code>	47
Shrink/stretch removed in <code>\FBthousandsep</code>	38	<code>\frquote</code> : <code>\FB@quotespace</code> (kern), changed into <code>\FB@guillspace</code>	29
<code>\datefrench</code> : Specific code for Plain finally removed (babel bug reported).	31		
v3.3c			
General: New command <code>\FBthousandsep</code> to customise numprint.	38		