

A Babel language definition file for French

frenchb.dtx v3.2f, 2017/01/15

Daniel Flipo
daniel.flipo@free.fr

Contents

1 The French language	2
1.1 Basic interface	2
1.2 Customisation	4
1.2.1 <code>\frenchbsetup</code>	5
1.2.2 Captions	8
1.3 Hyphenation checks	9
1.4 Changes	10
2 The code	12
2.1 Initial setup	12
2.2 Punctuation	15
2.2.1 Punctuation with LuaTeX	17
2.2.2 Punctuation with XeTeX	24
2.2.3 Punctuation with standard (pdf)TeX	27
2.2.4 Punctuation switches common to all engines	29
2.3 Commands for French quotation marks	30
2.4 Date in French	34
2.5 Extra utilities	35
2.6 Formatting numbers	39
2.7 Caption names	41
2.8 Dots...	46
2.9 More checks about packages' loading order	46
2.10 Setup options: keyval stuff	47
2.11 French lists	60
2.12 French indentation of sections	64
2.13 Formatting footnotes	65
2.14 Clean up and exit	68
3 Change History	69

1 The French language

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

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

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

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

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

Changes between version 3.0 and v3.2f are listed in subsection 1.4 p. 10.

An extensive documentation is available in French here:

<http://daniel.flipo.free.fr/frenchb>

1.1 Basic interface

In a multilingual document, some typographic rules are language dependent, i.e. spaces before ‘high punctuation’ (: ; ! ?) in French, others modify the general layout (i.e. layout of lists, footnotes, indentation of first paragraphs of sections) and should apply to the whole document.

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.

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*²:

1. the first paragraph of each section is indented (L^AT_EX only);
2. the default items in itemize environment are set to ‘—’ instead of ‘•’, and all vertical spacing and glue is deleted; it is possible to change ‘—’ to something else (‘-’ for instance) using `\frenchbsetup{}` (see section 1.2 p. 4);
3. vertical spacing in general L^AT_EX lists is shortened;
4. footnotes are displayed “à la française”.
5. the separator following the table or figure number in captions is printed as ‘-’ instead of ‘: ’; for changing this see 1.2.2 p. 8.

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

¹The file described in this section has version number v3.2f and was last revised on 2017/01/15.

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

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

1. French hyphenation patterns are made active;
2. ‘high punctuation’ characters (: ; ! ?) automatically add correct spacing in French; this is achieved using callbacks in Lua(La)TeX or ‘XeTeXinterchar’ mechanism in Xe(La)TeX; with TeX’82 and pdf(La)TeX these four characters are made active in the whole document;
3. `\today` prints the date in French;
4. the caption names are translated into French (L^AT_EX only). For customisation of caption names see section 1.2.2 p. 8.
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 commands `\og` and `\fg` which work in L^AT_EX 2_ε and PlainT_EX, their appearance depending on what is available to draw them; even if you use L^AT_EX 2_ε and T1-encoding, you should refrain from entering them as `<<~French quotation~>>`: `\og` and `\fg` provide better horizontal spacing (controlled by `\FBguillspace`). If French quote characters are available on your keyboard, you can use them, to get proper spacing in L^AT_EX 2_ε see option `og=«`, `fg=»` p. 8.

`\og` and `\fg` can be used outside French, they typeset then English quotes “ and ”.

A new command `\frquote{}` has been added in version 3.1 to enter French quotations. `\frquote{texte}` is equivalent to `\og texte \fg{}` for short quotations. For quotations spreading over more than one paragraph, `\frquote` will add at the beginning of every paragraph of the quotation either an opening French guillemet («), or a closing one (») depending on option `EveryParGuill=open` or `=close`, see p. 8.

`\frquote` is recommended to enter embedded quotations “à la française”, several variants are provided through options:

- with LuaTeX based engines, every line of the inner quotation will start with a French opening or closing guillemet (« or ») depending on option `EveryLineGuill=open` (default) or `=close` unless you explicitly set `EveryLineGuill=none`, then `\frquote{}` will behave as with non-LuaTeX engines;
- with all other engines, the inner quotation is surrounded by double quotes (“*texte*”) unless option `InnerGuillSingle=true`, then a) the inner quotation is printed as `< texte >` and b) if the inner quotation spreads over more than one paragraph, every paragraph included in the inner quotation starts with a `<` or a `>`, depending on option `EveryParGuill=open` or `close`.

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

2. A command `\up` is provided to typeset superscripts like `M\up{me}` (abbreviation for “Madame”), `1\up{er}` (for “premier”). Other commands are also provided for ordinals: `\ier`, `\iere`, `\iers`, `\ieres`, `\ieme`, `\iemes` (`3\iemes` prints 3^{es}). All these commands take advantage of real superscript letters when they are available in the current font.
3. Family names should be typeset in small capitals and never be hyphenated, the macro `\bsc` (boxed small caps) does this, e.g., `L.\bsc{Lamport}` will print the same as `L.\mbox{\textsc{Lamport}}`. Note that composed names (such as Dupont-Durant) may now be hyphenated on explicit hyphens, this differs from babel-french v. 1.x.
4. Commands `\primo`, `\secundo`, `\tertio` and `\quarto` print 1^o, 2^o, 3^o, 4^o. `\FrenchEnumerate{6}` prints 6^o.
5. Abbreviations for “Numéro(s)” and “numéro(s)” (`No` `Nos` `no` and `nos`) are obtained via the commands `\No`, `\Nos`, `\no`, `\nos`.
6. Two commands are provided to typeset the symbol for “degré”: `\degre` prints the raw character and `\degres` should be used to typeset temperatures (e.g., “20~\degres C” with an nobreak space), or for alcohols’ strengths (e.g., “45\degres” with *no* space in French).
7. In math mode the comma has to be surrounded with braces to avoid a spurious space being inserted after it, in decimal numbers for instance (see the T_EXbook p. 134). The command `\DecimalMathComma` makes the comma behave as an ordinary character *when the current language is French* (no space added); as a counterpart, if `\DecimalMathComma` is active, an explicit space has to be added in lists and intervals: `$(0,\ 1)$`, `$(x,\ y)$`. `\StandardMathComma` switches back to the standard behaviour of the comma in French.
The `icomma` package is an alternative workaround.
8. A command `\nombre` was provided in 1.x versions to easily format numbers in slices of three digits separated either by a comma in English or with a space in French; `\nombre` is now mapped to `\numprint` from `numprint.sty`, see `numprint.pdf` for more information.
9. babel-french has been designed to take advantage of the `xspace` package if present: adding `\usepackage{xspace}` in the preamble will force macros like `\fg`, `\ier`, `\ieme`, `\dots`, ..., to respect the spaces you type after them, for instance typing ‘`1\ier juin`’ will print ‘1^{er} juin’ (no need for a forced space after `1\ier`).

1.2 Customisation

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

1.2.1 `\frenchbsetup{options}`

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

The other options are listed below. Their default value is shown between braces, sometimes followed by a `'*'`. The `'*'` means that the default shown applies when `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. This option can be used to avoid conflicts with classes or packages which customise lists or footnotes.

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

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

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

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

`StandardItemizeEnv=true (false*)` ; `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 `false` reverts to the standard definition of `itemize`.

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

`StandardItemLabels=true (false*)` when set to `true` this option prevents `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 the example `\ding{43}` requires `\usepackage{pifont}`.

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

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

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

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

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

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

`FrenchFootnotes=false (true*)` reverts to the standard layout of footnotes. By default 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 thin space in the running text before the number or symbol calling the footnote. Making this option `false` reverts to the standard setting (no space added).

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

`AutoSpacePunctuation=false (true)` ; in French, the user *should* input a space before the four characters ‘:;!?’ but as many people forget about it (even among native French writers!), the default behaviour of babel-french is to automatically typeset nobreak 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 case⁴, so the default behaviour of of babel-french in that area should be fine in most circumstances.

⁴Unless option `OriginalTypewriter` is set, `\ttfamily` is redefined in French to switch off space tuning, see below.

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

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

`OriginalTypewriter=true (false)` prevents any customisation of `\ttfamily` and `\texttt{}` in French.

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

`PartNameFull=false (true)` ; when true, 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”.

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

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

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

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

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

`EveryParGuill=open, close, none (open)` ; sets whether an opening quote (`<<`) or a closing one (`>>`) or nothing should be printed by `\frquote{}` at the beginning of every paragraph in case of a level 1 (outer) quotation spreading

over more than one paragraph. This option is also considered for level 2 (inner) quotations to decide between < and > when `InnerGuillSingle=true` (see below).

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

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

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

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

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

1.2.2 Captions

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

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

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

Three new options are provided: if `CustomiseFigTabCaptions` is set to `false` the colon will be used as separator in all languages, with a proper space before the colon in French. The second option, `OldFigTabCaptions`, can be set to `true` to print figures' and tables' captions as they were with versions pre 3.0 of babel-french (using `\CaptionSeparator` in French and colon in other languages); this option only makes sense with the standard L^AT_EX classes `article`, `report` and `book`. The last option, `SmallCapsFigTabCaptions`, can be set to `false` to typeset `\figurename` and `\tablename` in French as "Figure" and "Table" rather than in small caps (the default).

1.3 Hyphenation checks

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

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

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

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

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

Frequent mismatches:

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

1.4 Changes

What's new in version 3.2?

The handling of footnotes has been redesigned for the beamer, memoir and koma-script classes. The layout of footnotes “à la française” should be unchanged but footnotes’ customisations offered by these classes (i.e. font or color changes) are now available even when option `FrenchFootnotes` is `true`.

A long standing bug regarding the xspace package has been fixed: `\xspace` has been moved up from the internal command `\FB@fg` to `\fg`; `\frquote{}` now works properly when the xspace package is loaded.

Version 3.2b is the first one designed to work with LuaTeX v. 0.95 as included in TeXLive 2016 (LuaTeX’s new glue node structure is not compatible with previous versions).

Warning to Lua(La)TeX users: starting with version 3.2b the lua code included in `frenchb.lua` will *not work* on older installations (TL2015 f.i.), so babel-french reverts to active characters while handling high punctuation with LuaTeX engines older than 0.95! The best way to go is to upgrade to TL2016 or equivalent asap. Xe(La)TeX and pdf(La)TeX users can safely use babel-french v. 3.2b and later on older installations too.

The internals of commands `\NoAutoSpacing`, `\ttfamilyFB`, `\rmfamilyFB` and `\sffamilyFB` have been completely redesigned in version 3.2c, they behave now consistently with all engines.

What's new in version 3.1?

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

What's new in version 3.0?

Many deep changes lead me to step babel-french’s version number to 3.0a:

- babel 3.9 is required now to process `frenchb.ldf`, this change allows for cleaner definitions of dates and captions for the Unicode engines LuaTeX and XeTeX and also provides a simpler syntax for end-users, see section 1.2.2 p.8.
- `\frenchbsetup{}` options management has been completely reworked; two new options added.
- Canadian French didn’t work as a normal babel’s dialect, it should now; btw. the French language should now be loaded as `french`, *not as frenchb* or `francais` and preferably as a *global* option of `\documentclass`. Some tolerance still exists in v3.0, but do not rely on it.
- babel-french no longer loads `frenchb.cfg`: customisation should definitely be done using `\frenchbsetup{}` options.
- Description lists labels are now indented; try setting `\descindentFB=0pt` (or `\listindentFB=0pt` for all lists) in the preamble if you don’t like it.

- The last but not least change affects the (recent) LuaTeX-based engines, (this means version 0.76 as included in TL2013 and up): active characters are no longer used in French for ‘high punctuation’⁵. Functionalities and user interface are unchanged.

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

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

Starting with version 3.0c, babel-french no longer customises lists with the beamer class and offers a new option (`INGuillSpace`) to follow French ‘Imprimerie Nationale’ recommendations regarding quotes’ spacing.

⁵The current babel-french version requires LuaTeX v. 0.95 as included in TL2016, see above.

2 The code

2.1 Initial setup

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

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

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

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

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

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

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

```

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

```

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

```

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

```

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

```

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

```

```
81 \bbl@tempa
```

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

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

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

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

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

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

We cannot rely on ε -TeX’s `\ifdefined` at this stage, as it is not defined in Plain T_EX format.

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

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

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

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

```

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

```

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

```

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

```

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

```

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

```

2.2 Punctuation

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

`\ifFB@active@punct`

```

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

```

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

With LuaTeX, starting with version 0.95, callbacks are used to get rid of active punctuation. With previous versions, ‘high punctuation’ characters remain active (see below).

```

130 \newif\ifFB@luatex@punct

```

```

131 \ifBLuaTeX
132   \ifnum\luatexversion<95
133     \ifx\PackageWarning\@undefined
134       \fb@warning{Please upgrade LuaTeX to version 0.95 or above!\\%
135         frenchb will make high punctuation characters (;:!) active\\%
136         with LuaTeX < 0.95.}%
137     \else
138       \PackageWarning{frenchb.ldf}{Please upgrade LuaTeX
139         to version 0.95 or above!\MessageBreak
140         frenchb will make high punctuation characters\MessageBreak
141         (;:!) active with LuaTeX < 0.95;\MessageBreak reported}%
142     \fi
143   \else
144     \FB@luatex@puncttrue\FB@active@punctfalse
145   \fi
146 \fi

```

\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 4095 instead of 255.

```

147 \newcount\FB@nonchar
148 \newif\ifFB@xetex@punct
149 \begingroup\expandafter\expandafter\expandafter\endgroup
150 \expandafter\ifx\csgname XeTeXinterchartokenstate\endcsname\relax
151 \else
152   \FB@xetex@puncttrue\FB@active@punctfalse
153   \ifdim\the\XeTeXversion\XeTeXrevision pt<0.99994pt
154     \FB@nonchar=255 \relax
155   \else
156     \FB@nonchar=4095 \relax
157   \fi
158 \fi

```

\FBcolonspace According to the I.N. specifications, the ‘:’ requires an inter-word space before it, **\FBthinspace** 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, both are user customisable.

```

159 \newcommand*{\FBcolonspace}{\space}
160 \newcommand*{\FBthinspace}{\hskip.5\fontdimen2\font \relax}

```

\FBcolonskip LuaTeX requires skips instead of commands, so we define `\FBcolonskip` and **\FBthinskip** `\FBthinskip` to hold the width/stretch/shrink specifications of `\FBcolonspace` and `\FBthinspace` for the `lrm10` font; these parameters will be scaled for the current font by the `frenchb.lua` script (see how p. 19). `\FBcolonskip` and `\FBthinskip` are also user customisable.

```

161 \newskip\FBcolonskip
162 \FBcolonskip=3.33pt plus 1.665pt minus 1.11pt \relax
163 \newskip\FBthinskip
164 \FBthinskip=1.6667pt \relax

```


With LuaTeX and XeTeX engines, babel-french handles French quotes together with ‘high punctuation’; 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. A matching attribute will be defined later for LuaTeX.

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

`\FB@spacing@off` Two internal commands to switch on and off all space tuning for all six characters `\FB@spacing@on` ‘;:!?«»’. They will be triggered by user command `\NoAutoSpacing` and by font family switching commands `\ttfamilyFB` `\rmfamilyFB` and `\sffamilyFB`. These four commands will now behave the same with any engine (up to version 3.2b, results were engine dependent).

```
166 \newcommand*{\FB@spacing@on}{%
167   \ifFB@luatex@punct
168     \FB@spacing=1 \relax
169   \else
170     \FB@spacingtrue
171   \fi}
172 \newcommand*{\FB@spacing@off}{%
173   \ifFB@luatex@punct
174     \FB@spacing=0 \relax
175   \else
176     \FB@spacingfalse
177   \fi}
```

2.2.1 Punctuation with LuaTeX

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

The following `\directlua` call ensures compatibility with LaTeX releases prior to 2015/10/01: the `\localleftbox` primitive⁶ introduced by Omega was prefixed with “luatex”, it should no longer be, see `ltnews23.tex` for details.

```
178 \ifFB@luatex@punct
179   \directlua{tex.enableprimitives("", tex.extraprimatives("omega"))}
```

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

```
180 \begingroup\expandafter\expandafter\expandafter\endgroup
181 \expandafter\ifx\csname newluafunction\endcsname\relax
```

This code is for Plain: `loadltnuatex.tex` if it hasn’t been loaded before babel.

```
182   \input ltnuatex.tex
183 \fi
```

`\FB@spacing=0` switches off any space tuning both before high punctuation characters and inside French quotes (i.e. function `french_punctuation` doesn’t alter the node list at all). `\FB@addDPspace=0` switches off automatic insertion of spaces before high punctuation characters (but typed spaces are still turned into nobreak thin- or word-spaces). `\FB@addGUILspace` will be set to 1 by option `og=«`, `fg=»`, thus enabling automatic insertion of proper spaces after ‘«’ and before ‘»’.

⁶used by `\frquote`, see p. 33.

```

184 \newattribute\FB@spacing      \FB@spacing=1 \relax
185 \newattribute\FB@addDPspace  \FB@addDPspace=1 \relax
186 \newattribute\FB@addGUILspace \FB@addGUILspace=0 \relax
187 \ifLaTeXe
188   \PackageInfo{frenchb.ldf}{No need for active punctuation
189     characters\MessageBreak with this version
190     of LuaTeX!\MessageBreak reported}
191 \else
192   \fb@info{No need for active punctuation characters\\
193     with this version of LuaTeX!}
194 \fi
195 \fi

```

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

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

```

196 local FB_punct_thin =
197   {[string.byte("!")] = true,
198    [string.byte("?")] = true,
199    [string.byte(";")] = true}
200 local FB_punct_thick =
201   {[string.byte(":")] = true}

```

Managing spacing after ‘«’ (U+00AB) and before ‘»’ (U+00BB) can be done by the way; we define two flags, FB_punct_left for characters requiring some space before them and FB_punct_right for ‘«’ which must be followed by some space. In case LuaTeX is used to output T1-encoded fonts instead of OpenType fonts, codes 0x13 and 0x14 have to be added for ‘«’ and ‘»’.

```

202 local FB_punct_left =
203   {[string.byte("!")] = true,
204    [string.byte("?")] = true,
205    [string.byte(";")] = true,
206    [string.byte(":")] = true,
207    [0x14] = true,
208    [0xBB] = true}
209 local FB_punct_right =
210   {[0x13] = true,
211    [0xAB] = true}

```

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

```

212 local FB_punct_null =
213   {[string.byte("!")] = true,
214    [string.byte("?")] = true,
215    [string.byte("[")] = true,
216    [string.byte("(")] = true,

```

or if the user has typed a nobreak space U+00A0 or a nobreak thin space U+202F before a ‘high punctuation’ character: no space should be added by babel-french. Same is true inside French quotes.

```

217   [0xA0] = true,

```

```

218 [0x202F] = true}
219 local FB_guil_null =
220 {[0xA0] = true,
221 [0x202F] = true}

```

Local definitions for nodes:

```

222 local new_node = node.new
223 local copy_node = node.copy
224 local node_id = node.id
225 local HLIST = node_id("hlist")
226 local TEMP = node_id("temp")
227 local KERN = node_id("kern")
228 local GLUE = node_id("glue")
229 local GLYPH = node_id("glyph")
230 local PENALTY = node_id("penalty")
231 local nobreak = new_node(PENALTY)
232 nobreak.penalty = 10000
233 local insert_node_before = node.insert_before
234 local insert_node_after = node.insert_after
235 local remove_node = node.remove

```

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

```

236 local thin10 = tex.skip['FBthinskip']
237 local thinwd = thin10.width/65536/3.33
238 local thinst = thin10.stretch/65536/1.665
239 local thinsh = thin10.shrink/65536/1.11
240 local coln10 = tex.skip['FBcolonskip']
241 local colnwd = coln10.width/65536/3.33
242 local colnst = coln10.stretch/65536/1.665
243 local colnsh = coln10.shrink/65536/1.11
244 local guil10 = tex.skip['FBguillskip']
245 local guilwd = guil10.width/65536/3.33
246 local guilst = guil10.stretch/65536/1.665
247 local guilsh = guil10.shrink/65536/1.11

```

and a function to scale them for the current font (beware of null values for fid, see \nullfont in TikZ, and of special fonts like lcircle1.pfb for which font.getfont(fid) does not return a proper font table, in such cases the function returns nil):

```

248 local font_table = {}
249 local function new_glue_scaled (fid,width,stretch,shrink)
250   if fid > 0 then
251     local fp = font_table[fid]
252     if not fp then
253       local ft = font.getfont(fid)
254       if ft then
255         font_table[fid] = ft.parameters
256         fp = font_table[fid]
257       end
258     end
259     local gl = new_node(GLUE,0)

```

```

260     if fp then
261         gl.width = width * fp.space
262         gl.stretch = stretch * fp.space_stretch
263         gl.shrink = shrink * fp.space_shrink
264         return gl
265     else
266         return nil
267     end
268 else
269     return nil
270 end
271 end

```

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

```

272 local FBspacing      = luatexbase.attributes['FB@spacing']
273 local addDPspace     = luatexbase.attributes['FB@addDPspace']
274 local addGUILspace   = luatexbase.attributes['FB@addGUILspace']
275 local has_attribute = node.has_attribute

```

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

```

276 local function french_punctuation (head)
277   for item in node.traverse_id(GLYPH, head) do
278     local lang = item.lang
279     local char = item.char
280     local fid  = item.font
281     local FRspacing = has_attribute(item, FBspacing)
282     FRspacing = FRspacing and FRspacing > 0
283     local SIG = has_attribute(item, addGUILspace)
284     SIG = SIG and SIG > 0
285     if lang == FR and FRspacing and
286        FB_punct_left[char] and fid > 0 then
287       local prev = item.prev
288       local prev_id, prev_subtype, prev_char
289       if prev then
290         prev_id = prev.id
291         prev_subtype = prev.subtype
292         if prev_id == GLYPH then
293           prev_char = prev.char
294         end
295       end

```

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

```

296     local is_glue = prev_id == GLUE
297     local glue_wd

```

```

298     if is_glue then
299         glue_wd = prev.width
300     end
301     local realglue = is_glue and glue_wd > 1

```

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

```

302     if FB_punct_thin[char] or FB_punct_thick[char] then
303         local SBDP = has_attribute(item, addDPspace)
304         local auto = SBDP and SBDP > 0
305         if auto then
306             if (prev_char and FB_punct_null[prev_char]) or
307                 (is_glue and glue_wd <= 1) or
308                 (prev_id == HLIST and prev_subtype == 3) or
309                 (prev_id == TEMP) then
310                 auto = false
311             end
312         end
313         local fbglue
314         if FB_punct_thick[char] then
315             fbglue = new_glue_scaled(fid,colnwd,colnst,colnsh)
316         else
317             fbglue = new_glue_scaled(fid,thinwd,thinst,thinsh)
318         end

```

In case `new_glue_scaled` fails (returns nil) the node list remains unchanged.

```

319         if (realglue or auto) and fbglue then
320             if realglue then
321                 head = remove_node(head,prev,true)
322             end
323             insert_node_before(head, item, copy_node(nobreak))
324             insert_node_before(head, item, copy_node(fbglue))
325         end

```

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

```

326         elseif SIG then

```

```

327         local addgl = (prev_char and not FB_guil_null[prev_char]) or
328             (not prev_char and
329                 prev_id ~= TEMP and
330                 not (prev_id == HLIST and prev_subtype == 3)
331             )

```

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

```

332         if is_glue and glue_wd <= 1 then
333             addgl = false
334         end
335         local fbglue = new_glue_scaled(fid,guilwd,guilst,guilsh)
336         if addgl and fbglue then
337             if is_glue then
338                 head = remove_node(head,prev,true)
339             end
340             insert_node_before(head, item, copy_node(nobreak))
341             insert_node_before(head, item, copy_node(fbglue))
342         end
343     end
344 end

```

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

```

345     if lang == FR and FRspacing and FB_punct_right[char]
346         and fid > 0 and SIG then
347         local next = item.next
348         local next_id, next_subtype, next_char, nextnext, kern_wd
349         if next then
350             next_id = next.id
351             next_subtype = next.subtype
352             if next_id == GLYPH then
353                 next_char = next.char

```

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

```

354         elseif next_id == KERN then
355             kern_wd = next.kern
356             if kern_wd == 0 then
357                 nextnext = next.next
358                 if nextnext then
359                     next = nextnext
360                     next_id = nextnext.id
361                     next_subtype = nextnext.subtype
362                     if next_id == GLYPH then
363                         next_char = nextnext.char
364                     end
365                 end
366             end

```

```

367         end
368     end
369     local is_glue = next_id == GLUE
370     if is_glue then
371         glue_wd = next.width
372     end
373     local addgl = (next_char and not FB_guil_null[next_char]) or
374                 (next and not next_char)

```

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

```

375     if is_glue and glue_wd == 0 then
376         addgl = false
377     end
378     local fid = item.font
379     local fbglue = new_glue_scaled(fid,guilwd,guilst,guilsh)
380     if addgl and fbglue then
381         if is_glue then
382             head = remove_node(head,next,true)
383         end
384         insert_node_after(head, item, copy_node(fbglue))
385         insert_node_after(head, item, copy_node(nobreak))
386     end
387 end
388 end
389 return head
390 end
391 return french_punctuation

```

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

```

392 \ifFB@luatex@punct
393   \newcommand*{\FB@luatex@punct@french}{%
394     \babel@save{\shorthandon}%
395     \babel@save{\shorthandoff}%
396     \def\shorthandoff##1{%
397       \ifx\PackageWarning\@undefined
398         \fb@warning{\noexpand\shorthandoff{;:!?} is helpless with
399           LuaTeX,\, use \noexpand\NoAutoSpacing
400           *inside a group* instead.}%
401       \else
402         \PackageWarning{frenchb.ldf}{\protect\shorthandoff{;:!?} is
403           helpless with LuaTeX,\MessageBreak use \protect\NoAutoSpacing
404           \space *inside a group* instead;\MessageBreak reported}%
405       \fi}%
406     \def\shorthandon##1{%
407   }
408   \FB@addto{extras}{\FB@luatex@punct@french}

```

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

```

409 \def\activate@luatexpunct{%
410   \directlua{%
411     FR = \the\l@french
412     local path = kpse.find_file("frenchb.lua", "lua")
413     if path then
414       local f = dofile(path)
415       luatexbase.add_to_callback("kerning",
416                                f, "frenchb.french_punctuation")
417     else
418       texio.write_nl('')
419       texio.write_nl('*****')
420       texio.write_nl('Error: frenchb.lua not found.')
421       texio.write_nl('*****')
422       texio.write_nl('')
423     end
424   }%
425 }
426 \fi

```

End of specific code for punctuation with LuaTeX engines.

2.2.2 Punctuation with XeTeX

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

The default value for `\XeTeXcharclass` is 0 for characters tokens and `\FB@nonchar` for all other tokens (glues, kerns, math and box boundaries, etc.). These defaults should not be changed otherwise the spacing before the ‘high punctuation’ characters and inside quotes might not be correct.

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

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

```

427 \ifFB@xetex@punct
428   \ifLaTeXe
429     \PackageInfo{frenchb.ldf}{No need for active punctuation characters%
430                  \MessageBreak with this version of XeTeX!%
431                  \MessageBreak reported}
432   \else
433     \fb@info{No need for active punctuation characters\\

```



```

434         with this version of XeTeX!}
435     \fi

```

Six new character classes are defined for babel-french.

```

436     \newXeTeXintercharclass\FB@punctthick
437     \newXeTeXintercharclass\FB@punctthin
438     \newXeTeXintercharclass\FB@punctnul
439     \newXeTeXintercharclass\FB@guilo
440     \newXeTeXintercharclass\FB@guilf
441     \newXeTeXintercharclass\FB@guilnul

```

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

```

442     \def\FBsavevariable@loop#1#2{\begingroup
443         \toks@\expandafter{\originalTeX #1}%
444         \edef\x{\endgroup
445             \def\noexpand\originalTeX{\the\toks@ #2=\the#1#2\relax}}%
446         \x}

```

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

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

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

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

```

447     \def\FB@charlist{"21,"3A,"3B,"3F,"AB,"BB,"28,"5B,"A0,"202F,%
448                     "29,"5D,"7B,"7D,"2C,"2D,"2E,"22,"25,"27,"60,"2019}

```

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

```

449     \newcommand*{\FB@xetex@punct@french}{%
450         \babel@savevariable{\XeTeXinterchartokenstate}%
451         \babel@save{\shorthandon}%
452         \babel@save{\shorthandoff}%
453         \bbl@for\FB@char\FB@charlist
454             {\FBsavevariable@loop{\XeTeXcharclass}{\FB@char}}%
455         \def\shorthandoff##1{%
456             \ifx\PackageWarning\@undefined
457                 \fb@warning{\noexpand\shorthandoff{;!?} is helpless with
458                     XeTeX,\ use \noexpand\NoAutoSpacing
459                     *inside a group* instead.}%
460             \else
461                 \PackageWarning{frenchb.ldf}{\protect\shorthandoff{;!?} is
462                     helpless with XeTeX,\MessageBreak use \protect\NoAutoSpacing
463                     \space *inside a group* instead;\MessageBreak reported}%
464             \fi}%
465         \def\shorthandon##1{%

```

Let's now set the classes 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).

```

466 \XeTeXinterchartokenstate=1
467 \XeTeXcharclass '\: = \FB@punctthick
468 \XeTeXinterchartoks \z@ \FB@punctthick = {%
469   \ifFB@spacing\ifhmode\FDP@colonspace\fi\fi}%
470 \XeTeXinterchartoks \FB@guilf \FB@punctthick = {%
471   \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: unfortunately `\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.

```

472 \XeTeXinterchartoks \FB@nonchar \FB@punctthick = {%
473   \ifFB@spacing
474     \ifhmode
475       \ifdim\lastskip>1sp
476         \unskip\penalty\@M\FBcolonspace
477       \else
478         \FDP@colonspace
479       \fi
480     \fi
481   \fi}%
482 \bbl@for\FB@char
483   {\';,\!\,'?}%
484   {\XeTeXcharclass\FB@char=\FB@punctthin}%
485 \XeTeXinterchartoks \z@ \FB@punctthin = {%
486   \ifFB@spacing\ifhmode\FDP@thinspace\fi\fi}%
487 \XeTeXinterchartoks \FB@guilf \FB@punctthin = {%
488   \ifFB@spacing\FDP@thinspace\fi}%
489 \XeTeXinterchartoks \FB@nonchar \FB@punctthin = {%
490   \ifFB@spacing
491     \ifhmode
492       \ifdim\lastskip>1sp
493         \unskip\penalty\@M\FBthinspace
494       \else
495         \FDP@thinspace
496       \fi
497     \fi
498   \fi}%
499 \XeTeXinterchartoks \FB@guilo \z@ = {%
500   \ifFB@spacing\FBguillspace\fi}%
501 \XeTeXinterchartoks \FB@guilo \FB@nonchar = {%
502   \ifFB@spacing\FBguillspace\ignorespaces\fi}%
503 \XeTeXinterchartoks \z@ \FB@guilf = {%
504   \ifFB@spacing\FBguillspace\fi}%
505 \XeTeXinterchartoks \FB@punctthin \FB@guilf = {%
506   \ifFB@spacing\FBguillspace\fi}%

```

```

507 \XeTeXinterchartoks \FB@nonchar \FB@guilf = {%
508     \ifFB@spacing\unskip\FBguillspace\fi}%

```

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

```

509      \bbl@for\FB@char
510      {\ '[ , '\ ( , "A0 , "202F}%
511      {\XeTeXcharclass\FB@char=\FB@punctnul}%

```

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

```

512 \bbl@for\FB@char
513 {\'{, \'., \'-, \'}, \'}, \'%, "22,"27,"60,"2019}%
514 {\XeTeXcharclass\FB@char=\zeta}%
515 }
516 \FB@addto{extras}{\FB@xetex@punct@french}

```

End of specific code for punctuation with modern XeTeX engines.

517 \fi

2.2.3 Punctuation with standard (pdf)TeX

In standard (pdf)TeX we need to make the four characters ; ! ? and : ‘active’ and provide their definitions.

```
518 \ifFB@active@punct
519 \initiate@active@char{:}%
520 \initiate@active@char{;}%
521 \initiate@active@char{!}%
522 \initiate@active@char{?}%
```

We first tune the amount of space before ; ! ? and :. This should only happen in horizontal mode, hence the test `\ifhmode`.

In horizontal mode, if a space has been typed before ‘;’ we remove it and put an unbreakable `\FBthinspace` instead. If no space has been typed, we add `\FDP@thinspace` which will be defined, up to the user’s wishes, as `\FBthinspace`, or as `\@empty`.

```

523 \declare@shorthand{french}{;}{;%
524 \ifFB@spacing
525 \ifhmode
526 \ifdim\lastskip>1sp
527 \unskip\penalty\@M\FBthinspace
528 \else
529 \FDP@thinspace
530 \fi
531 \fi
532 \fi

```

Now we can insert a ; character.

```
533     \string;}
```

The next three definitions are very similar.

```

534 \declare@shorthand{french}{!}{%
535   \ifFB@spacing
536   \ifhmode
537     \ifdim\lastskip>lsp
538       \unskip\penalty\@M\FBthinspace
539     \else
540       \FDP@thinspace
541     \fi
542   \fi
543 \fi
544 \string!}
545 \declare@shorthand{french}{?}{%
546   \ifFB@spacing
547   \ifhmode
548     \ifdim\lastskip>lsp
549       \unskip\penalty\@M\FBthinspace
550     \else
551       \FDP@thinspace
552     \fi
553   \fi
554 \fi
555 \string?}
556 \declare@shorthand{french}{:}{%
557   \ifFB@spacing
558   \ifhmode
559     \ifdim\lastskip>lsp
560       \unskip\penalty\@M\FBcolonspace
561     \else
562       \FDP@colonspace
563     \fi
564   \fi
565 \fi
566 \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.

```

567 \declare@shorthand{system}{:}{\string:}
568 \declare@shorthand{system}{!}{\string!}
569 \declare@shorthand{system}{?}{\string?}
570 \declare@shorthand{system}{;}{\string;}
571 %}

```

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

```

572 \FB@addto{extras}{\languageshorthands{french}%

```

These characters are ‘turned on’ once, later their definition may vary. Don’t misunderstand the following code: they keep being active all along the document, even when leaving French.

```

573 \bbl@activate{:}\bbl@activate{;}%

```

```

574 \bbl@activate{!}\bbl@activate{?}%
575 }
576 \FB@addto{noextras}{%
577 \bbl@deactivate{:}\bbl@deactivate{;}%
578 \bbl@deactivate{!}\bbl@deactivate{?}%
579 }
580 \fi

```

2.2.4 Punctuation switches common to all engines

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

```

581 \newif\ifFBAutoSpacePunctuation \FBAutoSpacePunctuationtrue

```

`\AutoSpaceBeforeFDP` `\autospace@beforeFDP` and `\noautospace@beforeFDP` are internal commands. `\NoAutoSpaceBeforeFDP` `\autospace@beforeFDP` defines `\FDP@thinspace` and `\FDP@colonspace` as unbreakable spaces and sets LuaTeX attribute `\FB@addDPspace` to 1 (true), while `\noautospace@beforeFDP` lets these spaces empty and sets flag `\FB@addDPspace` to 0 (false). User commands `\AutoSpaceBeforeFDP` and `\NoAutoSpaceBeforeFDP` do the same and take care of the flag `\ifFBAutoSpacePunctuation` in \LaTeX . Set the default now for Plain (done later for \LaTeX).

```

582 \def\autospace@beforeFDP{%
583 \ifFB@luatex@punct\FB@addDPspace=1 \fi
584 \def\FDP@thinspace{\penalty\@M\FBthinspace}%
585 \def\FDP@colonspace{\penalty\@M\FBcolonspace}}
586 \def\noautospace@beforeFDP{%
587 \ifFB@luatex@punct\FB@addDPspace=0 \fi
588 \let\FDP@thinspace\@empty
589 \let\FDP@colonspace\@empty}
590 \ifLaTeXe
591 \def\AutoSpaceBeforeFDP{\autospace@beforeFDP
592 \FBAutoSpacePunctuationtrue}
593 \def\NoAutoSpaceBeforeFDP{\noautospace@beforeFDP
594 \FBAutoSpacePunctuationfalse}
595 \AtEndOfPackage{\AutoSpaceBeforeFDP}
596 \else
597 \let\AutoSpaceBeforeFDP\autospace@beforeFDP
598 \let\NoAutoSpaceBeforeFDP\noautospace@beforeFDP
599 \AutoSpaceBeforeFDP
600 \fi

```

`\rmfamilyFB` In $\LaTeX_{2\epsilon}$ `\ttfamily` (and hence `\texttt`) will be redefined ‘AtBeginDocument’ `\sffamilyFB` as `\ttfamilyFB` so that no space is added before the four ; : ! ? characters, `\ttfamilyFB` even if `AutoSpacePunctuation` is true. When `AutoSpacePunctuation` is false, the eventually typed spaces are left unchanged (not turned into thin spaces, no penalty added). `\rmfamily` and `\sffamily` need to be redefined also (`\ttfamily` is not always used inside a group, its effect can be cancelled by `\rmfamily` or `\sffamily`).

These redefinitions can be canceled if necessary, for instance to recompile older documents, see option [OriginalTypewriter](#) below.

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

```
601 \ifLaTeXe
602   \DeclareRobustCommand\ttfamilyFB{\FB@spacing@off \ttfamilyORI}
603   \DeclareRobustCommand\rmfamilyFB{\FB@spacing@on  \rmfamilyORI}
604   \DeclareRobustCommand\sffamilyFB{\FB@spacing@on  \sffamilyORI}
605 \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, LuaTeX and XeTeX based engines) and is meant to be used inside a group.

```
606 \DeclareRobustCommand*\NoAutoSpacing{%
607   \FB@spacing@off
608   \ifFB@active@punct\shorthandoff{;:!?}\fi
609 }
```

2.3 Commands for French quotation marks

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

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

```
610 \ifLaTeXe
611 \else
612   \ifFBunicode
613     \def\guillemotleft{{\char"00AB}}
614     \def\guillemotright{{\char"00BB}}
615     \def\textquotedblleft{{\char"201C}}
616     \def\textquotedblright{{\char"201D}}
617   \else
618     \def\guillemotleft{\leavevmode\raise0.25ex
619       \hbox{$\scriptscriptstyle\ll$}}
620     \def\guillemotright{\raise0.25ex
621       \hbox{$\scriptscriptstyle\gg$}}
622     \def\textquotedblleft{‘}
623     \def\textquotedblright{’}
624   \fi
625   \let\xspace\relax
626 \fi
```

`\FB@og` The next step is to provide correct spacing after `\guillemotleft` and before `\guillemotright`: a space precedes and follows quotation marks but no line break is allowed neither *after* the opening one, nor *before* the closing one. `\FBguillspace` which does the spacing, has been fine tuned by Thierry Bouche to 80% of an inter-word space but with reduced stretchability. French quotes (including spacing) are printed by `\FB@og` and `\FB@fg`, the expansion of the top level commands `\og` and `\og` is different in and outside French.

LuaTeX which requires skips; `\FBguillskip` is computed from `\FBguillspace` for the `lrm10` font, its dimensions will be scaled by `frenchb.lua` for the current font and used after ‘`«`’ and before ‘`»`’ when option `og=«`, `fg=»` is set.

```
627 \newskip\FBguillskip
628 \FBguillskip=2.664pt plus 0.500pt minus 0.888pt \relax
629 \newcommand*{\FBguillspace}{\penalty\@M\hskip.8\fontdimen2\font
630                               plus.3\fontdimen3\font
631                               minus.8\fontdimen4\font}
```

For efficiency reasons, `\FB@og` and `\FB@fg` rely on `\FBguillspace` with LuaTeX engines (`\FB@spacing` is set to 0 locally).

```
632 \ifFB@luatex@punct
633   \DeclareRobustCommand*{\FB@og}{\leavevmode
634     \bgroup\FB@spacing=0 \guillemotleft\egroup
635     \FBguillspace}
636   \DeclareRobustCommand*{\FB@fg}{\ifdim\lastskip>\z@\unskip\fi
637     \FBguillspace
638     \bgroup\FB@spacing=0 \guillemotright\egroup}
639 \fi
```

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

```
640 \ifFB@xetex@punct
641   \DeclareRobustCommand*{\FB@og}{\leavevmode
642     \bgroup\FB@spacingfalse\guillemotleft\egroup
643     \FBguillspace}
644   \DeclareRobustCommand*{\FB@fg}{\ifdim\lastskip>\z@\unskip\fi
645     \FBguillspace
646     \bgroup\FB@spacingfalse\guillemotright\egroup}
647 \fi
648 \ifFB@active@punct
649   \DeclareRobustCommand*{\FB@og}{\leavevmode
650     \guillemotleft
651     \FBguillspace}
652   \DeclareRobustCommand*{\FB@fg}{\ifdim\lastskip>\z@\unskip\fi
653     \FBguillspace
654     \guillemotright}
655 \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.

```

656 \newcommand*\og{}\@empty}
657 \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).

```

658 \ifLaTeXe
659   \def\bbl@frenchguillemets{\renewcommand*\og{}\FB@og}%
660                               \renewcommand*\fg{}\FB@fg\xspace}}
661   \renewcommand*\og{}\textquotedblleft}
662   \renewcommand*\fg{}\ifdim\lastskip>\z@ \unskip\fi
663                               \textquotedblright\xspace}
664 \else
665   \def\bbl@frenchguillemets{\let\og\FB@og
666                               \let\fg\FB@fg}
667   \def\og{\textquotedblleft}
668   \def\fg{\ifdim\lastskip>\z@ \unskip\fi \textquotedblright}
669 \fi

670 \FB@addto{extras}{\babel@save\og \babel@save\fg \bbl@frenchguillemets}

```

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

```

671 \newcommand*\ogi{}\FB@og}
672 \newcommand*\fgi{}\FB@fg}
673 \newcommand*\ogii{\textquotedblleft}
674 \newcommand*\fgii{\textquotedblright}

and the needed technical stuff to handle options:

675 \newcount\FBguill@level
676 \newtoks\FB@everypar
677 \newif\ifFBcloseguill \FBcloseguilltrue
678 \newif\ifFBInnerGuillSingle
679 \def\FBguillopen{\bgroup\NoAutoSpacing\guillemotleft\egroup}
680 \def\FBguillclose{\bgroup\NoAutoSpacing\guillemotright\egroup}
681 \let\FBguillnone\relax
682 \let\FBeveryparguill\FBguillopen
683 \ifFB@luatex@punct
684   \let\FBeverylanguill\FBguillopen
685 \else
686   \let\FBeverylanguill\FBguillnone
687 \fi

```

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

```

688 \ifLaTeXe

```



```

689 \DeclareRobustCommand\frquote{%
690     \@ifstar{\FBcloseguillfalse\frquote}%
691     {\FBcloseguilltrue\frquote}}
692 \else
693 \newcommand\frquote[1]{\frquote{#1}}
694 \fi

```

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

```

695 \newcommand{\frquote}[1]{%
696     \leavevmode
697     \advance\FBguill@level by \@ne

```

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

```

698 \def\FB@quotespace{\kern.8\fontdimen2\font}%
699 \ifcase\FBguill@level
700 \or

```

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

```

701 \FB@everypar=\everypar
702 \ifx\FBeveryparguill\relax
703 \else
704 \def\FBeverypar@quote{\FBeveryparguill\FB@quotespace}%
705 \everypar=\expandafter{\the\everypar \FBeverypar@quote}%
706 \fi
707 \ogi #1\fgi
708 \or

```

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

```

709 \ifx\FBverylineguill\FBguillopen
710 \localleftbox{\guillemotleft\FB@quotespace}%
711 \let\FBeverypar@quote\relax
712 \ogi #1\ifFBcloseguill\fgi\fi
713 \else
714 \ifx\FBverylineguill\FBguillclose
715 \localleftbox{\guillemotright\FB@quotespace}%
716 \let\FBeverypar@quote\relax
717 \ogi #1\ifFBcloseguill\fgi\fi
718 \else

```

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

```

719 \let\FBeverypar@quote\relax
720 \ifFBInnerGuillSingle
721 \def\ogii{\leavevmode
722     \guilsinglleft\FBguillspace}%
723 \def\fgii{\ifdim\lastskip>\z@\unskip\fi
724     \FBguillspace\guilsinglright}%
725 \ifx\FBeveryparguill\FBguillopen

```

```

726         \def\FBeverypar@quote{\guilsinglleft\FB@quotespace}%
727         \fi
728         \ifx\FBeveryparguill\FBguillclose
729         \def\FBeverypar@quote{\guilsinglright\FB@quotespace}%
730         \fi
731     \fi
732     \ogii #1\ifFBcloseguill \fgii \fi
733 \fi
734 \fi
735 \else
Warn if \FBguill@level  $\geq 3$ :
736     \ifx\PackageWarning\undefined
737     \fb@warning{\noexpand\frquote\space accepts no more than
738         two levels.\\ Quotation not printed.}%
739     \else
740     \PackageWarning{frenchb.ldf}{%
741         \protect\frquote\space accepts no more than two levels
742         \MessageBreak Quotation not printed. Reported}
743     \fi
744 \fi
Clean on exit: adjust \FBguill@level and restore \localleftbox and \everypar.
745 \advance\FBguill@level by \m@ne
746 \ifx\FBeverylanguill\FBguillnone\else\localleftbox{}\fi
747 \ifx\FBeveryparguill\relax\else\everypar=\FB@everypar\fi
748 }

```

2.4 Date in French

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

```

749 \ifLaTeXe
750 \def\BabelLanguages{french,acadian}
751 \StartBabelCommands*\BabelLanguages\{date}
752     [unicode, fontenc=EU1 EU2, charset=utf8]
753     \SetString\monthiiname{février}
754     \SetString\monthviiiname{août}
755     \SetString\monthxiiname{décembre}
756 \StartBabelCommands*\BabelLanguages\{date}
757     \SetStringLoop{month#lname}{%
758         janvier,f\'evrier,mars,avril,mai,juin,juillet,%
759         ao\^ut,septembre,octobre,novembre,d\'ecembre}
760     \SetString\today{{\number\day}\ifnum1=\day {\ier}\fi\space
761         \csname month\romannumeral\month name\endcsname \space
762         \number\year
763     }
764 \EndBabelCommands
765 \else

```

```

766 \ifBUnicode
767   \namedef{date\CurrentOption}{%
768     \def\today{{\number\day}\ifnum1=\day {\ier}\fi \space
769       \ifcase\month
770         \or janvier\or février\or mars\or avril\or mai\or
771         juin\or juillet\or août\or septembre\or
772         octobre\or novembre\or décembre\fi
773       \space \number\year}}
774 \else
775   \@namedef{date\CurrentOption}{%
776     \def\today{{\number\day}\ifnum1=\day {\ier}\fi \space
777       \ifcase\month
778         \or janvier\or f'evrier\or mars\or avril\or mai\or
779         juin\or juillet\or ao^ut\or septembre\or
780         octobre\or novembre\or d'ecembre\fi
781       \space \number\year}}
782 \fi
783 \fi

```

2.5 Extra utilities

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

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

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

```

784 \newif\ifFB@poorman
785 \newdimen\FB@Mht
786 \ifLaTeXe
787   \AtEndOfPackage{\RequirePackage{scalefnt}}

```

\FB@up@fake holds the definition of fake superscripts. The scaling ratio is 0.65, raising is computed to put the top of lower case letters (like 'm') just under the top of upper case letters (like 'M'), precisely 12% down. The chosen settings look correct for most fonts, but can be tuned by the end-user if necessary by changing \FBsupR and \FBsupS commands.

\FB@lc is defined as \MakeLowercase to inhibit the uppercasing of superscripts (this may happen in page headers with the standard classes but is wrong); \FB@lc can be redefined to do nothing by option LowercaseSuperscripts=false of \frenchbsetup{ }.

```

788 \newcommand*{\FBsupR}{-0.12}

```

```

789 \newcommand*{\FBsupS}{0.65}
790 \newcommand*{\FB@lc}[1]{\MakeLowercase{#1}}
791 \DeclareRobustCommand*{\FB@up@fake}[1]{%
792   \settoheight{\FB@Mht}{M}%
793   \addtolength{\FB@Mht}{\FBsupR \FB@Mht}%
794   \addtolength{\FB@Mht}{-\FBsupS ex}%
795   \raisebox{\FB@Mht}{\scalefont{\FBsupS}{\FB@lc{#1}}}%
796 }

```

The only packages I currently know to take advantage of real superscripts are a) realscripts used in conjunction with XeLaTeX or LuaLaTeX and OpenType fonts having the font feature ‘VerticalPosition=Superior’ and b) fourier (from version 1.6) when Expert Utopia fonts are available.

\FB@up checks whether the current font is a Type1 ‘Expert’ (or ‘Pro’) font with real superscripts or not (the code works currently only with fourier-1.6 but could work with any Expert Type1 font with built-in superscripts, see below), and decides to use real or fake superscripts. It works as follows: the content of \f@family (family name of the current font) is split by \FB@split into two pieces, the first three characters (‘fut’ for Fourier, ‘ppl’ for Adobe’s Palatino, ...) stored in \FB@firstthree and the rest stored in \FB@suffix which is expected to be ‘x’ or ‘j’ for expert fonts.

```

797 \def\FB@split#1#2#3#4\@nil{\def\FB@firstthree{#1#2#3}%
798   \def\FB@suffix{#4}}
799 \def\FB@x{x}
800 \def\FB@j{j}
801 \DeclareRobustCommand*{\FB@up}[1]{%
802   \bgroup \FB@poormantrue
803   \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.

```

804   \edef\reserved@a{\lowercase{%
805     \noexpand\IfFileExists{\f@encoding\FB@firstthree -sup.fd}}}%
806   \reserved@a
807   {\ifx\FB@suffix\FB@x \FB@poormanfalse\fi
808    \ifx\FB@suffix\FB@j \FB@poormanfalse\fi
809    \ifFB@poorman \FB@up@fake{#1}%
810    \else \FB@up@real{#1}%
811    \fi}%
812   {\FB@up@fake{#1}}%
813   \egroup}

```

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

```

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

```

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

```

816 \DeclareRobustCommand*{\fup}[1]{%

```

```

817 \ifx\realsuperscript\@undefined
818 \FB@up{#1}%
819 \else
820 \bgroup\let\fakesuperscript\FB@up@fake
821 \realsuperscript{\FB@lc{#1}}\egroup
822 \fi}

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

823 \providecommand*\up{\relax}

Poor man's definition of \up for Plain.

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

```

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

```

\ier 827 \def\ieme{\up{e}\xspace}
\iere 828 \def\iemes{\up{es}\xspace}
\iemes 829 \def\ier{\up{er}\xspace}
\iers 830 \def\iers{\up{ers}\xspace}
\ieres 831 \def\iere{\up{re}\xspace}
      832 \def\ieres{\up{res}\xspace}

```

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

```

\no 833 \newcommand*\FrenchEnumerate[1]{%
\nos 834 \up{o}\kern+.3em}
\nos 835 \newcommand*\FrenchPopularEnumerate[1]{%
\primo 836 \up{o})\kern+.3em}
\frimo) Typing \primo should result in '1°',

```

```

837 \def\primo{\FrenchEnumerate1}
838 \def\secundo{\FrenchEnumerate2}
839 \def\tertio{\FrenchEnumerate3}
840 \def\quarto{\FrenchEnumerate4}

```

while typing \frimo) gives '1°'.

```

841 \def\frimo{\FrenchPopularEnumerate1}
842 \def\fsecundo{\FrenchPopularEnumerate2}
843 \def\ftertio{\FrenchPopularEnumerate3}
844 \def\fquarto{\FrenchPopularEnumerate4}

```

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

```

845 \DeclareRobustCommand*\No{\N\up{o}\kern+.2em}
846 \DeclareRobustCommand*\no{\n\up{o}\kern+.2em}
847 \DeclareRobustCommand*\Nos{\N\up{os}\kern+.2em}
848 \DeclareRobustCommand*\nos{\n\up{os}\kern+.2em}

```

\bsc As family names should be written in small capitals and never be hyphenated, we provide a command (its name comes from Boxed Small Caps) to input them easily. Note that this command has changed with version 2 of babel-french: a \kern0pt is used instead of \hbox because \hbox would break microtype's font expansion;

as a (positive?) side effect, composed names (such as Dupont-Durand) can now be hyphenated on explicit hyphens. Usage: Jean~\bsc{Duchemin}.

```
849 \DeclareRobustCommand*\bsc}[1]{\leavevmode\beginngroup\kern0pt
850                                     \scshape #1\endgroup}
851 \ifLaTeXe\else\let\scshape\relax\fi
```

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

```
852 \ifFBunicode
853   \newcommand*\at{{\char"0040}}
854   \newcommand*\circonflexe{{\char"005E}}
855   \newcommand*\tild{{\char"007E}}
856   \newcommand*\boi{{\textbackslash}}
857   \newcommand*\degree{{\char"00B0}}
858 \else
859   \ifLaTeXe
860     \DeclareTextSymbol\at{T1}{64}
861     \DeclareTextSymbol\circonflexe{T1}{94}
862     \DeclareTextSymbol\tild{T1}{126}
863     \DeclareTextSymbolDefault\at{T1}
864     \DeclareTextSymbolDefault\circonflexe{T1}
865     \DeclareTextSymbolDefault\tild{T1}
866     \DeclareRobustCommand*\boi{{\textbackslash}}
867     \DeclareRobustCommand*\degree{{\r{}}}
868   \else
869     \def\T@one{T1}
870     \ifx\fontencoding\T@one
871       \newcommand*\degree{{\char6}}
872     \else
873       \newcommand*\degree{{\char23}}
874     \fi
875     \newcommand*\at{{\char64}}
876     \newcommand*\circonflexe{{\char94}}
877     \newcommand*\tild{{\char126}}
878     \newcommand*\boi{{\backslash}}
879   \fi
880 \fi
```

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

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

```
881 \ifLaTeXe
```

```

882 \newcommand*{\degrees}{\degree}
883 \ifFBunicode
884   \DeclareRobustCommand*{\degrees}{\degree}
885 \else
886   \def\Warning@degree@TSone{%
887     \PackageWarning{frenchb.1df}{%
888       Degrees would look better in TS1-encoding:%
889       \MessageBreak add \protect
890       \usepackage{textcomp} to the preamble.%
891       \MessageBreak Degrees used}}
892   \AtBeginDocument{\ifx\DeclareEncodingSubset\@undefined
893     \DeclareRobustCommand*{\degrees}{%
894       \leavevmode\hbox to 0.3em{\hss\degree\hss}%
895       \Warning@degree@TSone
896       \global\let\Warning@degree@TSone\relax}%
897   \else
898     \DeclareRobustCommand*{\degrees}{%
899       \hbox{\UseTextSymbol{TS1}{\textdegree}}}%
900   \fi
901 }
902 \fi
903 \else
904   \newcommand*{\degrees}{%
905     \leavevmode\hbox to 0.3em{\hss\degree\hss}}
906 \fi

```

2.6 Formatting numbers

\StandardMathComma As mentioned in the T_EXbook 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.

```

907 \newif\ifFB@icomma
908 \newcount\mc@charclass
909 \newcount\mc@charfam
910 \newcount\mc@charslot
911 \newcount\std@mcc
912 \newcount\dec@mcc
913 \ifBLaTeX
914   \mc@charclass=\Umathcharclass'\,
915   \newcommand*{\dec@math@comma}{%
916     \mc@charfam=\Umathcharfam'\,
917     \mc@charslot=\Umathcharslot'\,
918     \Umathcode'\,= 0 \mc@charfam \mc@charslot
919   }
920   \newcommand*{\std@math@comma}{%
921     \mc@charfam=\Umathcharfam'\,

```

```

922 \mc@charslot=\Umathcharslot'\,
923 \Umathcode'\,=\ \mc@charclass \mc@charfam \mc@charslot
924 }
925 \else
926 \std@mcc=\mathcode'\,
927 \dec@mcc=\std@mcc
928 \@tempcnta=\std@mcc
929 \divide\@tempcnta by "1000
930 \multiply\@tempcnta by "1000
931 \advance\dec@mcc by -\@tempcnta
932 \newcommand*\dec@math@comma{\mathcode'\,=\dec@mcc}
933 \newcommand*\std@math@comma{\mathcode'\,=\std@mcc}
934 \fi
935 \newcommand*\DecimalMathComma{%
936 \iflanguage{french}{\dec@math@comma}{}%
937 \ifFB@icomma\else\FB@addto{extras}{\dec@math@comma}\fi
938 }
939 \newcommand*\StandardMathComma{%
940 \std@math@comma
941 \ifFB@icomma\else\FB@addto{extras}{\std@math@comma}\fi
942 }
943 \ifLaTeXe
944 \AtBeginDocument{\@ifpackageloaded{icomma}%
945 \FB@icommatrue}%
946 \FB@addto{noextras}{\std@math@comma}}%
947 }
948 \else
949 \FB@addto{noextras}{\std@math@comma}
950 \fi

```

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

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

```

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

```

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

```

953 \let\FBstop@here\relax
954 \def\FBclean@on@exit{\let\ifLaTeXe\undefined
955 \let\LaTeXetrue\undefined
956 \let\LaTeXefalse\undefined}
957 \ifx\magnification\@undefined
958 \else
959 \def\FBstop@here{\ifFB@luatex@punct
960 \activate@luatexpunct

```



```

961 \fi
962 \FBcleanon@exit
963 \ldf@quit\CurrentOption\endinput}
964 \fi
965 \FBstop@here

```

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

```

966 \renewcommand*{\nombre}[1]{\Warning@nombre{#1}}
967 \newcommand*{\Warning@nombre}[1]{%
968   \ifdefined\numprint
969     \numprint{#1}%
970   \else
971     \PackageWarning{frenchb.ldf}{%
972       \protect\nombre\space now relies on package numprint.sty,%
973       \MessageBreak add \protect
974       \usepackage[autolanguage]{numprint},\MessageBreak
975       see file numprint.pdf for more options.\MessageBreak
976       \protect\nombre\space called}%
977     \global\let\Warning@nombre\relax
978     {#1}%
979   \fi
980 }

```

2.7 Caption names

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

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

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

```

981 \ifx\FBfigtabshape\undefined \let\FBfigtabshape\scshape \fi

```

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

```

982 \StartBabelCommands*{\BabelLanguages}{captions}
983   [unicode, fontenc=EU1 EU2, charset=utf8]
984   \SetString{\refname}{Références}
985   \SetString{\abstractname}{Résumé}
986   \SetString{\prefacename}{Préface}
987   \SetString{\contentsname}{Table des matières}
988   \SetString{\ccname}{Copie à }
989   \SetString{\proofname}{Démonstration}
990   \SetStringLoop{ordinal#1}{%
991     Première,Deuxième,Troisième,Quatrième,Cinquième,%
992     Sixième,Septième,Huitième,Neuvième,Dixième,Onzième,%

```

```

993      Douzième,Treizième,Quatorzième,Quinzième,Seizième,%
994      Dix-septième,Dix-huitième,Dix-neuvième,Vingtième}
995 \StartBabelCommands*{\BabelLanguages}{captions}
996   \SetString{\refname}{R\’ef\’erences}
997   \SetString{\abstractname}{R\’esum\’e}
998   \SetString{\bibname}{Bibliographie}
999   \SetString{\prefacename}{Pr\’eface}
1000  \SetString{\chaptername}{Chapitre}
1001  \SetString{\appendixname}{Annexe}
1002  \SetString{\contentsname}{Table des mati\’eres}
1003  \SetString{\listfigurename}{Table des figures}
1004  \SetString{\listtablename}{Liste des tableaux}
1005  \SetString{\indexname}{Index}
1006  \SetString{\figurename}{{\FBfigtabshape Figure}}
1007  \SetString{\tablename}{{\FBfigtabshape Table}}
1008  \SetString{\pagename}{page}
1009  \SetString{\seename}{voir}
1010  \SetString{\alsoname}{voir aussi}
1011  \SetString{\enclname}{P.~J. }
1012  \SetString{\ccname}{Copie \’a }
1013  \SetString{\headtoname}{}
1014  \SetString{\proofname}{D\’emonstration}
1015  \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`.

```

1016  \SetStringLoop{ordinal#1}{%
1017      Premi\’ere,Deuxi\’eme,Troisi\’eme,Quatri\’eme,Cinqui\’eme,%
1018      Sixi\’eme,Septi\’eme,Huiti\’eme,Neuvi\’eme,Dixi\’eme,Onzi\’eme,%
1019      Douzi\’eme,Treizi\’eme,Quatorzi\’eme,Quinzi\’eme,Seizi\’eme,%
1020      Dix-septi\’eme,Dix-huiti\’eme,Dix-neuvi\’eme,Vingti\’eme}
1021  \AfterBabelCommands{%
1022      \DeclareRobustCommand*{\FB@emptypart}{\def\thepart{}}%
1023      \DeclareRobustCommand*{\FB@partname}{%
1024          \ifFBPartNameFull
1025              \csname ordinal\romannumeral\value{part}\endcsname\space
1026              partie\FB@emptypart
1027          \else
1028              Partie%
1029          \fi}%
1030      }
1031  \SetString{\partname}{{\FB@partname}}
1032 \EndBabelCommands

```

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

```

1033 \AtEndOfPackage{%
1034   \ifFB@koma

```

```

1035     \ifdefined\partformat
1036         \FB@addto{captions}{%
1037             \ifFBPartNameFull
1038                 \babel@save\partformat
1039                 \renewcommand*{\partformat}{\partname}%
1040             \fi}%
1041     \fi
1042 \fi
1043 }

```

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

\CaptionSeparator Let's consider now captions in figures and tables. In French, captions in figures and tables should never be printed as 'Figure 1:' which is the default in standard $\text{\LaTeX} 2_{\epsilon}$ classes; the ':' is made active too late, no space is added before it. With LuaLaTeX and 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 $\text{\LaTeX} 2_{\epsilon}$ according to Frank Mittelbach), is saved in `\STD@makecaption`. 'AtBeginDocument' we compare it to its current definition (some classes like `memoir`, `koma-script` classes, `AMS` classes, `ua-thesis.cls`... change it). If they are identical, babel-french just adds a hook called `\FBCaption@Separator` to `\@makecaption`; `\FBCaption@Separator` defaults to ':' as in the standard `\@makecaption` and will be changed to ' :' in French 'AtBeginDocument'; it can be also set to `\CaptionSeparator` (' - ') using [CustomiseFigTabCaptions](#). While saving the standard definition of `\@makecaption` we have to make sure that characters ':' and '>' have `\catcode 12` (babel-french makes ':' active and `spanish.ldf` makes '>' active).

```

1044 \bgroup
1045 \catcode': =12 \catcode'> =12 \relax
1046 \long\gdef\STD@makecaption#1#2{%
1047     \vskip\abovecaptionskip
1048     \sbox\@tempboxa{#1: #2}%
1049     \ifdim \wd\@tempboxa >\hsize
1050         #1: #2\par
1051     \else
1052         \global \@minipagefalse
1053         \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
1054     \fi
1055     \vskip\belowcaptionskip}
1056 \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.

```

1057 \newif\if@FBwarning@capsep
1058 \@FBwarning@capseptrue
1059 \newcommand{\FBWarning}[2]{\PackageWarning{#1}{#2}}
1060 \newcommand*\CaptionSeparator{\space\textendash\space}
1061 \def\FBCaption@Separator{: }
1062 \long\def\FB@makecaption#1#2{%
1063   \vskip\abovecaptionskip
1064   \sbox\@tempboxa{#1\FBCaption@Separator #2}%
1065   \ifdim \wd\@tempboxa >\hsize
1066     #1\FBCaption@Separator #2\par
1067   \else
1068     \global \@minipagefalse
1069     \hbext\@hsize{\hfil\box\@tempboxa\hfil}%
1070   \fi
1071   \vskip\belowcaptionskip}

```

Disable the standard warning with AMS and SMF classes.

```

1072 \@ifclassloaded{amsart}{\@FBwarning@capsepfalse}{}
1073 \@ifclassloaded{amsbook}{\@FBwarning@capsepfalse}{}
1074 \@ifclassloaded{amstex}{\@FBwarning@capsepfalse}{}
1075 \@ifclassloaded{amslatex}{\@FBwarning@capsepfalse}{}
1076 \@ifclassloaded{amproc}{\@FBwarning@capsepfalse}{}
1077 \@ifclassloaded{smfart}{\@FBwarning@capsepfalse}{}
1078 \@ifclassloaded{smfbook}{\@FBwarning@capsepfalse}{}

```

Disable the standard warning unless high punctuation is active.

```

1079 \ifFB@active@punct\else\@FBwarning@capsepfalse\fi

```

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

```

1080 \newif\ifFB@koma
1081 \@ifclassloaded{memoir}{\@FBwarning@capsepfalse}{}
1082 \@ifclassloaded{scrartcl}{\@FBwarning@capsepfalse\FB@komatrue}{}
1083 \@ifclassloaded{scrbook}{\@FBwarning@capsepfalse\FB@komatrue}{}
1084 \@ifclassloaded{scrreprt}{\@FBwarning@capsepfalse\FB@komatrue}{}

```

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

```

1085 \@ifclassloaded{beamer}{\@FBwarning@capsepfalse}{}
1086 \ifdefined\@makecaption\else\@FBwarning@capsepfalse\fi

```

The caption, subcaption and floatrow packages are compatible with babel-french if they are loaded after babel.

Check if package caption is loaded now (before babel/babel-french), then issue a warning advising to load it after babel/babel-french and disable the standard warning.

```

1087 \@ifpackageloaded{caption}
1088   {\FBWarning{frenchb.1df}}%

```

```

1089      {Please load the "caption" package\MessageBreak
1090      AFTER babel/frenchb; reported}%
1091      \@FBwarning@capsepfalse}%
1092      {}

```

Same for package subcaption.

```

1093 \@ifpackageloaded{subcaption}
1094   {\FBWarning{frenchb.ldf}%
1095    {Please load the "subcaption" package\MessageBreak
1096    AFTER babel/frenchb; reported}%
1097    \@FBwarning@capsepfalse}%
1098   {}

```

Same for package floatrow.

```

1099 \@ifpackageloaded{floatrow}
1100   {\FBWarning{frenchb.ldf}%
1101    {Please load the "floatrow" package\MessageBreak
1102    AFTER babel/frenchb; reported}%
1103    \@FBwarning@capsepfalse}%
1104   {}

```

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

```

1105 \@AtBeginDocument{%
1106   \ifx\@makecaption\STD@makecaption
1107     \global\let\@makecaption\FB@makecaption

```

Do not overwrite \FBCaption@Separator if already saved as ‘:’ for other languages and set to \CaptionSeparator by \extrasfrench when French is the main language.

```

1108   \ifFBoldFigTabCaptions
1109   \else
1110     \def\FBCaption@Separator{{\autospace@beforeFDP : }}%
1111     \fi
1112     \ifFBCustomiseFigTabCaptions
1113       \ifx\bbl@main@language\FB@french
1114         \def\FBCaption@Separator{\CaptionSeparator}%
1115       \fi
1116     \fi
1117     \@FBwarning@capsepfalse
1118   \fi
1119   \if@FBwarning@capsep
1120     \FBWarning{frenchb.ldf}%
1121     {Figures’ and tables’ captions might look like\MessageBreak
1122     ‘Figure 1:’ which is wrong in French.\MessageBreak
1123     Check your class or packages to change this;\MessageBreak
1124     reported}%
1125   \fi
1126   \let\FB@makecaption\relax
1127   \let\STD@makecaption\relax

```

2.8 Dots...

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

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

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

```
1129 \ifFBunicode
1130   \let\FBtextellipsis\textellipsis
1131 \else
1132   \DeclareTextSymbol{\FBtextellipsis}{LY1}{133}
1133   \DeclareTextCommandDefault{\FBtextellipsis}{%
1134     .\kern\fontdimen3\font.\kern\fontdimen3\font.\xspace}
1135 \fi
```

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

```
1136 \newcommand*{\Tdots@}{\@xp\textellipsis}
1137 \newcommand*{\Mdots@}{\@xp\mdots@}
1138 \AtBeginDocument{\DeclareRobustCommand*{\dots}{\relax
1139   \csname\ifmmode M\else T\fi dots@\endcsname}%
1140   \ifdefined\@xp\else\let\@xp\relax\fi
1141   \ifdefined\mdots@\else\let\Mdots@\mathellipsis\fi
1142 }
1143 \def\bbl@frenchdots{\babel@save\Tdots@ \let\Tdots@\FBtextellipsis}
1144 \FB@addto{extras}{\bbl@frenchdots}
```

2.9 More checks about packages' loading order

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

```
1145 \ifFB@active@punct
1146   \@ifpackageloaded{listings}
1147     {\FBWarning{frenchb.ldf}%
1148       {Please load the "listings" package\MessageBreak
1149         AFTER babel/frenchb; reported}%
1150     }{}
1151 \fi
```

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

```

1152 \newif\if@FBwarning@natbib
1153 \ifFB@active@punct
1154   \@ifpackageloaded{natbib}{\@FBwarning@natbibtrue}
1155 \fi
1156 \AtBeginDocument{%
1157   \if@FBwarning@natbib
1158     \@ifpackageloaded{natbib}{\@FBwarning@natbibfalse}%
1159   \fi
1160   \if@FBwarning@natbib
1161     \FBWarning{frenchb.ldf}%
1162     {Please load the "natbib" package\MessageBreak
1163      BEFORE babel/frenchb; reported}%
1164   \fi
1165 }

```

2.10 Setup options: keyval stuff

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

Option processing can occur either in `\frenchbsetup{}`, but *only for options explicitly set* by `\frenchbsetup{}`, or ‘AtBeginDocument’; any option affecting `\extrasfrench{}` *must* be processed by `\frenchbsetup{}`: when French is the main language, `\extrasfrench{}` is executed by babel when it switches the main language and this occurs *before* reading the stuff postponed by babel-french ‘AtBeginDocument’. Reexecuting `\extrasfrench{}` is an option which was used up to v2.6h, it has been dropped in v3.0a because of its side-effects (f.i. `\babel@save` and `\babel@savevariable` did not work for French).

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

```

1166 \newcommand*{\frenchbsetup}[1]{%
1167   \setkeys{FB}{#1}%
1168 }%
1169 \@onlypreamble\frenchbsetup

```

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

```

1170 \newif\ifFBShowOptions           \FBShowOptionsfalse
1171 \newif\ifFBStandardLayout        \FBStandardLayouttrue
1172 \newif\ifFBGlobalLayoutFrench    \FBGlobalLayoutFrenchtrue
1173 \newif\ifFBReduceListSpacing     \FBReduceListSpacingfalse
1174 \newif\ifFBListOldLayout         \FBListOldLayoutfalse
1175 \newif\ifFBCompactItemize        \FBCompactItemizefalse
1176 \newif\ifFBStandardItemizeEnv    \FBStandardItemizeEnvtrue
1177 \newif\ifFBStandardEnumerateEnv  \FBStandardEnumerateEnvtrue

```

```

1178 \newif\ifFBStandardItemLabels \FBStandardItemLabelstrue
1179 \newif\ifFBStandardLists \FBStandardListstrue
1180 \newif\ifFBIndentFirst \FBIndentFirstfalse
1181 \newif\ifFBFrenchFootnotes \FBFrenchFootnotesfalse
1182 \newif\ifFBAutoSpaceFootnotes \FBAutoSpaceFootnotesfalse
1183 \newif\ifFBOriginalTypewriter \FBOriginalTypewriterfalse
1184 \newif\ifFBThinColonSpace \FBThinColonSpacefalse
1185 \newif\ifFBThinSpaceInFrenchNumbers \FBThinSpaceInFrenchNumbersfalse
1186 \newif\ifFBFrenchSuperscripts \FBFrenchSuperscriptstrue
1187 \newif\ifFBLowercaseSuperscripts \FBLowercaseSuperscriptstrue
1188 \newif\ifFBPartNameFull \FBPartNameFulltrue
1189 \newif\ifFBCustomiseFigTabCaptions \FBCustomiseFigTabCaptionsfalse
1190 \newif\ifFBOldFigTabCaptions \FBOldFigTabCaptionsfalse
1191 \newif\ifFBSmallCapsFigTabCaptions \FBSmallCapsFigTabCaptionstrue
1192 \newif\ifFBSuppressWarning \FBSuppressWarningfalse
1193 \newif\ifFBINGuillSpace \FBINGuillSpacefalse

```

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

```

1194 \edef\FB@french{\CurrentOption}
1195 \AtEndOfPackage{%
1196   \ifx\bbl@main@language\FB@french
1197     \FBGlobalLayoutFrenchtrue
1198     \@ifclassloaded{beamer}%
1199       {\PackageInfo{frenchb.ldf}{%
1200         No list customisation for the beamer class,%
1201         \MessageBreak reported}}%
1202       {\FBReduceListSpacingtrue
1203        \FBStandardItemizeEnvfalse
1204        \FBStandardEnumerateEnvfalse
1205        \FBStandardItemLabelsfalse}%
1206     \FBIndentFirsttrue
1207     \FBFrenchFootnotesttrue
1208     \FBAutoSpaceFootnotesttrue
1209     \FBCustomiseFigTabCaptionstrue
1210   \else
1211     \FBGlobalLayoutFrenchfalse
1212   \fi

```

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

```

1213 \RequirePackage{keyval}%
1214 \define@key{FB}{ShowOptions}[true]%
1215   {\csname FBShowOptions#1\endcsname}%

```



```

1216 \define@key{FB}{StandardLayout}[true]%
1217         {\csname FBStandardLayout#1\endcsname
1218         \ifFBStandardLayout
1219             \FBReduceListSpacingfalse
1220             \FBStandardItemizeEnvtrue
1221             \FBStandardItemLabelstrue
1222             \FBStandardEnumerateEnvtrue
1223             \FBIndentFirstfalse
1224             \FBFrenchFootnotesfalse
1225             \FBAutoSpaceFootnotesfalse
1226             \FBGlobalLayoutFrenchfalse
1227         \else
1228             \FBReduceListSpacingtrue
1229             \FBStandardItemizeEnvfalse
1230             \FBStandardItemLabelsfalse
1231             \FBStandardEnumerateEnvfalse
1232             \FBIndentFirsttrue
1233             \FBFrenchFootnotesttrue
1234             \FBAutoSpaceFootnotesttrue
1235         \fi}%
1236 \define@key{FB}{GlobalLayoutFrench}[true]%
1237         {\csname FBGlobalLayoutFrench#1\endcsname

```

If this key is set to **true** when French is the main language, nothing to do: all flags keep their default value. If this key is set to **false**, nothing to do either: `\babel@save` will do the job.

```

1238         \ifFBGlobalLayoutFrench
1239             \ifx\bbbl@main@language\FB@french
1240             \else
1241                 \PackageWarning{frenchb.ldb}%
1242                 {Option 'GlobalLayoutFrench' skipped:%
1243                 \MessageBreak French is *not*
1244                 babel's last option.\MessageBreak}%
1245             \fi
1246         \fi}%
1247 \define@key{FB}{ReduceListSpacing}[true]%
1248         {\csname FBReduceListSpacing#1\endcsname}%
1249 \define@key{FB}{ListOldLayout}[true]%
1250         {\csname FBListOldLayout#1\endcsname
1251         \ifFBListOldLayout
1252             \FBStandardEnumerateEnvtrue
1253             \renewcommand*{\FrenchLabelItem}{\textendash}%
1254         \fi}%
1255 \define@key{FB}{CompactItemize}[true]%
1256         {\csname FBCompactItemize#1\endcsname
1257         \ifFBCompactItemize
1258             \FBStandardItemizeEnvfalse
1259             \FBStandardEnumerateEnvfalse
1260         \else
1261             \FBStandardItemizeEnvtrue
1262             \FBStandardEnumerateEnvtrue

```

```

1263         \fi}%
1264 \define@key{FB}{StandardItemizeEnv}[true]%
1265         {\csname FBStandardItemizeEnv#1\endcsname}%
1266 \define@key{FB}{StandardEnumerateEnv}[true]%
1267         {\csname FBStandardEnumerateEnv#1\endcsname}%
1268 \define@key{FB}{StandardItemLabels}[true]%
1269         {\csname FBStandardItemLabels#1\endcsname}%
1270 \define@key{FB}{ItemLabels}{%
1271     \renewcommand*{\FrenchLabelItem}{#1}}%
1272 \define@key{FB}{ItemLabeli}{%
1273     \renewcommand*{\Frlabelitemi}{#1}}%
1274 \define@key{FB}{ItemLabelii}{%
1275     \renewcommand*{\Frlabelitemii}{#1}}%
1276 \define@key{FB}{ItemLabeliii}{%
1277     \renewcommand*{\Frlabelitemiii}{#1}}%
1278 \define@key{FB}{ItemLabeliv}{%
1279     \renewcommand*{\Frlabelitemiv}{#1}}%
1280 \define@key{FB}{StandardLists}[true]%
1281         {\csname FBStandardLists#1\endcsname
1282         \ifFBStandardLists
1283             \FBReduceListSpacingfalse
1284             \FBCompactItemizefalse
1285             \FBStandardItemizeEnvtrue
1286             \FBStandardEnumerateEnvtrue
1287             \FBStandardItemLabelstrue
1288         \else
1289             \FBReduceListSpacingtrue
1290             \FBCompactItemizetrue
1291             \FBStandardItemizeEnvfalse
1292             \FBStandardEnumerateEnvfalse
1293             \FBStandardItemLabelsfalse
1294         \fi}%
1295 \define@key{FB}{IndentFirst}[true]%
1296         {\csname FBIndentFirst#1\endcsname}%
1297 \define@key{FB}{FrenchFootnotes}[true]%
1298         {\csname FBFrenchFootnotes#1\endcsname}%
1299 \define@key{FB}{AutoSpaceFootnotes}[true]%
1300         {\csname FBAutoSpaceFootnotes#1\endcsname}%
1301 \define@key{FB}{AutoSpacePunctuation}[true]%
1302         {\csname FBAutoSpacePunctuation#1\endcsname}%
1303 \define@key{FB}{OriginalTypewriter}[true]%
1304         {\csname FBOriginalTypewriter#1\endcsname}%
1305 \define@key{FB}{ThinColonSpace}[true]%
1306         {\csname FBThinColonSpace#1\endcsname}%
1307 \define@key{FB}{ThinSpaceInFrenchNumbers}[true]%
1308         {\csname FBThinSpaceInFrenchNumbers#1\endcsname}%
1309 \define@key{FB}{FrenchSuperscripts}[true]%
1310         {\csname FBFrenchSuperscripts#1\endcsname}%
1311 \define@key{FB}{LowercaseSuperscripts}[true]%
1312         {\csname FBLowercaseSuperscripts#1\endcsname}%
1313 \define@key{FB}{PartNameFull}[true]%

```

```

1314         {\csname FBPartNameFull#1\endcsname}%
1315 \define@key{FB}{CustomiseFigTabCaptions}[true]%
1316         {\csname FBCustomiseFigTabCaptions#1\endcsname}%
1317 \define@key{FB}{OldFigTabCaptions}[true]%
1318         {\csname FBOldFigTabCaptions#1\endcsname
\CurrentOption no longer defined. It's value has been saved in \FB@CurOpt while
reading frenchb.ldf.
1319         \ifFBOldFigTabCaptions
1320         \FB@addto{extras}{\babel@save\FBCaption@Separator
1321         \def\FBCaption@Separator{\CaptionSeparator}}%
1322         \fi}%
1323 \define@key{FB}{SmallCapsFigTabCaptions}[true]%
1324         {\csname FBSmallCapsFigTabCaptions#1\endcsname
1325         \ifFBSmallCapsFigTabCaptions
1326         \let\FBfigtabshape\scshape
1327         \else
1328         \let\FBfigtabshape\relax
1329         \fi}%
1330 \define@key{FB}{SuppressWarning}[true]%
1331         {\csname FBSuppressWarning#1\endcsname
1332         \ifFBSuppressWarning
1333         \renewcommand{\FBWarning}[2]{\relax}%
1334         \fi}%

```

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

```

1335 \define@key{FB}{INGuillSpace}[true]%
1336         {\csname FBINGuillSpace#1\endcsname}%
1337 \define@key{FB}{InnerGuillSingle}[true]%
1338         {\csname FBInnerGuillSingle#1\endcsname}%
1339 \define@key{FB}{EveryParGuill}{\expandafter\let\expandafter
1340         \FBeveryparguill\csname FBguill#1\endcsname}%
1341 \define@key{FB}{EveryLineGuill}{\expandafter\let\expandafter
1342         \FBeverylineguill\csname FBguill#1\endcsname
1343         \ifFB@luatex@punct
1344         \else
1345         \let\FBeverylineguill\FBguillnone
1346         \PackageWarning{frenchb.ldf}%
1347         {Option 'EveryLineGuill' skipped:%
1348         \MessageBreak this option is for
1349         LuaTeX *only*.\MessageBreak Reported}%
1350         \fi}%

```

Inputing French quotes as *single characters* when they are available on the keyboard (through a compose key for instance) is more comfortable than typing `\og` and `\fg`. With pdfTeX (or old LuaTeX and XeTeX engines), quote characters are made active and expand to `\og\ignorespaces` and `{\fg}` respectively if the current language is French, and to `\guillemotleft` and `\guillemotright` otherwise (think of German quotes), this is done by `\FB@@og` and `\FB@@fg`; thus correct unbreakable spaces will be added automatically to French quotes. The quote characters typed in depend on the input encoding, it can be single-byte (latin1, latin9, applemac,...) or multi-bytes

(utf-8, utf8x); the inputenc package has to be loaded before the `\begin{document}` with the proper coding option, so we check if `\DeclareInputText` is defined. Life is much simpler here with modern LuaTeX or XeTeX engines: we just have to activate the `\FB@addGUIILspace` attribute for LuaTeX or set `\XeTeXcharclass` of quotes to the proper value for XeTeX.

```
1351 \define@key{FB}{og}{%
1352     \ifFBunicode
```

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

```
1353     \ifFB@luatex@punct
1354     \FB@addGUIILspace=1 \relax
1355 \fi
```

then with XeTeX it is a bit more tricky:

```
1356     \ifFB@xetex@punct
```

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

```
1357         \XeTeXcharclass"13 = \FB@guilo
1358         \XeTeXcharclass"AB = \FB@guilo
1359         \XeTeXcharclass"A0 = \FB@guilnul
1360         \XeTeXcharclass"202F = \FB@guilnul
1361     \fi
```

Issue a warning with older Unicode engines requiring active characters.

```
1362     \ifFB@active@punct
1363     \FBWarning{frenchb.ldf}%
1364         {Option og=« not supported with this version
1365         of\MessageBreak LuaTeX/XeTeX; reported}%
1366 \fi
1367 \else
```

This is for conventional TeX engines:

```
1368     \newcommand*{\FB@@og}{%
1369         \iflanguage{french}%
1370         {\ifFB@spacing\FB@og\ignorespaces
1371         \else\guillemotleft
1372         \fi}%
1373         {\guillemotleft}}%
1374 \AtBeginDocument{%
1375     \ifdefined\DeclareInputText
1376     \ifdefined\uc@dclc
```

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

```
1377         \uc@dclc{171}{default}}{\FB@@og}%
1378 \else
```

if encoding is not utf8x, try utf8...

```
1379     \ifdefined\DeclareUnicodeCharacter
```

```

utf8 loaded, use \DeclareUnicodeCharacter,
1380         \DeclareUnicodeCharacter{00AB}{\FB@og}%
1381     \else
        if utf8 is not loaded either, we assume 8-bit character input encoding. Package
        MULEenc (from CJK) defines \mule@def to map characters to control sequences.
1382         \@tempcnta'#1\relax
1383         \ifdefined\mule@def
1384             \mule@def{11}{\FB@og}%
1385         \else
1386             \DeclareInputText{\the\@tempcnta}{\FB@og}%
1387         \fi
1388     \fi
1389 \fi
1390 \else
        Package inputenc not loaded, no way...
1391     \PackageWarning{frenchb.ldb}%
1392     {Option 'og' requires package inputenc.\MessageBreak}%
1393 \fi
1394 }%
1395 \fi
1396 }%

Same code for the closing quote.
1397 \define@key{FB}{fg}{%
1398     \ifFBunicode
1399         \ifFB@luatex@punct
1400             \FB@addGUILspace=1 \relax
1401         \fi
1402         \ifFB@xetex@punct
1403             \XeTeXcharclass"14 = \FB@guilf
1404             \XeTeXcharclass"BB = \FB@guilf
1405             \XeTeXcharclass"A0 = \FB@guilnul
1406             \XeTeXcharclass"202F = \FB@guilnul
1407         \fi
1408         \ifFB@active@punct
1409             \FBWarning{frenchb.ldb}%
1410             {Option fg=> not supported with this version
1411             of\MessageBreak LuaTeX/XeTeX; reported}%
1412         \fi
1413     \else
1414         \newcommand*{\FB@fg}{%
1415             \iflanguage{french}%
1416                 {\ifFB@spacing\FB@fg
1417                 \else\guillemotright
1418                 \fi}%
1419             {\guillemotright}}%
1420     \AtBeginDocument{%
1421         \ifdefined\DeclareInputText
1422         \ifdefined\uc@clc
1423             \uc@clc{187}{default}{\FB@fg}%

```

```

1424         \else
1425             \ifdefined\DeclareUnicodeCharacter
1426                 \DeclareUnicodeCharacter{00BB}{\FB@fg}%
1427             \else
1428                 \@tempcnta'#1\relax
1429                 \ifdefined\mule@def
1430                     \mule@def{27}{\FB@fg}%
1431                 \else
1432                     \DeclareInputText{\the\@tempcnta}{\FB@fg}%
1433                 \fi
1434             \fi
1435         \fi
1436     \else
1437         \PackageWarning{frenchb.ldf}%
1438             {Option 'fg' requires package inputenc.\MessageBreak}%
1439     \fi
1440 }%
1441 \fi
1442 }%
1443 }

```

\FBprocess@options \FBprocess@options will be executed at \begin{document}: it first checks about packages loaded in the preamble (possibly after babel) which customise lists: currently enumitem, paralist and enumerate; then it processes the options as set by \frenchbsetup{} or forced for compatibility with packages loaded in the preamble. When French is the main language, \extrasfrench and \captionsfrench *have already been processed* by babel at \begin{document} *before* \FBprocess@options.

```

1444 \newcommand*{\FBprocess@options}{%

```

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

```

1445 \ifpackageloaded{enumitem}{%
1446     \ifFBStandardItemizeEnv
1447     \else
1448         \FBStandardItemizeEnvtrue
1449         \PackageInfo{frenchb.ldf}%
1450             {Setting StandardItemizeEnv=true for\MessageBreak
1451                 compatibility with enumitem package,\MessageBreak}%
1452     \fi
1453     \ifFBStandardEnumerateEnv
1454     \else
1455         \FBStandardEnumerateEnvtrue
1456         \PackageInfo{frenchb.ldf}%
1457             {Setting StandardEnumerateEnv=true for\MessageBreak
1458                 compatibility with enumitem package,\MessageBreak}%
1459     \fi}}%
1460 \ifpackageloaded{paralist}{%
1461     \ifFBStandardItemizeEnv
1462     \else
1463         \FBStandardItemizeEnvtrue
1464         \PackageInfo{frenchb.ldf}%

```

```

1465         {Setting StandardItemizeEnv=true for\MessageBreak
1466         compatibility with paralist package,\MessageBreak}%
1467     \fi
1468     \ifFBStandardEnumerateEnv
1469     \else
1470         \FBStandardEnumerateEnvtrue
1471         \PackageInfo{frenchb.ldb}%
1472         {Setting StandardEnumerateEnv=true for\MessageBreak
1473         compatibility with paralist package,\MessageBreak}%
1474     \fi}}}%
1475 \@ifpackageloaded{enumerate}{%
1476     \ifFBStandardEnumerateEnv
1477     \else
1478         \FBStandardEnumerateEnvtrue
1479         \PackageInfo{frenchb.ldb}%
1480         {Setting StandardEnumerateEnv=true for\MessageBreak
1481         compatibility with enumerate package,\MessageBreak}%
1482     \fi}}}%

```

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

```

1483 \def\FB@ufl{\update@frenchlists}
1484 \ifx\bbl@main@language\FB@french
1485     \update@frenchlists
1486 \fi

```

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

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

```

1487 \ifFBAutoSpacePunctuation
1488     \autospace@beforeFDP
1489 \else
1490     \noautospace@beforeFDP
1491 \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.

```

1492 \ifFBOriginalTypewriter
1493 \else
1494     \let\ttfamilyORI\ttfamily
1495     \let\rmfamilyORI\rmfamily
1496     \let\sffamilyORI\sffamily
1497     \let\ttfamily\ttfamilyFB
1498     \let\rmfamily\rmfamilyFB
1499     \let\sffamily\sffamilyFB
1500 \fi

```

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

```

1501 \ifFBThinColonSpace
1502 \ifFB@luatex@punct
1503 \FBcolonskip=\FBthinskip\relax
1504 \else
1505 \renewcommand*{\FBcolonspace}{\FBthinspace}%
1506 \fi
1507 \fi

```

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

```

1508 \ifFBINGuillSpace
1509 \ifFB@luatex@punct
1510 \FBguillskip=3.33pt plus 1.665pt minus 1.11pt \relax
1511 \else
1512 \renewcommand*{\FBguillspace}{\space}%
1513 \fi
1514 \fi

```

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

```

1515 \@ifpackageloaded{numprint}%
1516 {\ifnprt@autolanguage
1517 \providecommand*{\npstylefrench}{}%
1518 \ifFBThinSpaceInFrenchNumbers
1519 \renewcommand*\npstylefrench{%
1520 \npthousandsep{\,}%
1521 \npdecimalsign{,}%
1522 \npproductsign{\cdot}%
1523 \npunitseparator{\,}%
1524 \npdegreeseperator{}}%
1525 \nppercentseparator{\nprt@unitsep}%
1526 }%
1527 \else
1528 \renewcommand*\npstylefrench{%
1529 \npthousandsep{~}%
1530 \npdecimalsign{,}%
1531 \npproductsign{\cdot}%
1532 \npunitseparator{\,}%
1533 \npdegreeseperator{}}%
1534 \nppercentseparator{\nprt@unitsep}%
1535 }%
1536 \fi
1537 \npaddtolanguage{french}{french}%
1538 \fi}{}%

```

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


```

1539 \ifBFrenchSuperscripts
1540   \DeclareRobustCommand*\up{\@ifstar{\FB@up@fake}{\fup}}%
1541 \else
1542   \DeclareRobustCommand*\up{\@ifstar{\FB@up@fake}%
1543                                   {\textsuperscript}}%
1544 \fi

```

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

```

1545 \ifBLLowercaseSuperscripts
1546 \else
1547   \renewcommand*\FB@lc[1]{##1}%
1548 \fi

```

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

```

1549 \ifBFCustomiseFigTabCaptions
1550   \ifFB@koma
1551     \renewcommand*\captionformat{\CaptionSeparator}%
1552   \fi
1553   \@ifclassloaded{memoir}%
1554     {\captiondelim{\CaptionSeparator}}{}%
1555   \@ifclassloaded{beamer}%
1556     {\defbeamertemplate{caption label separator}{FBcustom}{%
1557       \CaptionSeparator}%
1558     \setbeamertemplate{caption label separator}{FBcustom}}{}%
1559 \else

```

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

```

1560   \ifFB@koma
1561     \renewcommand*\captionformat{{\autospace@beforeFDP : }}%
1562   \fi
1563   \@ifclassloaded{memoir}%
1564     {\captiondelim{{\autospace@beforeFDP : }}%
1565     }{}%
1566   \@ifclassloaded{beamer}%
1567     {\defbeamertemplate{caption label separator}{FBcolon}{%
1568       {\autospace@beforeFDP : }}%
1569     \setbeamertemplate{caption label separator}{FBcolon}%
1570     }{}%
1571 \fi

```

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

```

1572 \ifBFSHOWOptions
1573   \GenericWarning{* }{%
1574     * **** List of possible options for frenchb ****\MessageBreak
1575     [Default values between brackets when frenchb is loaded *LAST*]%
1576     \MessageBreak
1577     ShowOptions=true [false]\MessageBreak
1578     StandardLayout=true [false]\MessageBreak
1579     GlobalLayoutFrench=false [true]\MessageBreak

```

```

1580 StandardLists=true [false]\MessageBreak
1581 IndentFirst=false [true]\MessageBreak
1582 ReduceListSpacing=false [true]\MessageBreak
1583 ListOldLayout=true [false]\MessageBreak
1584 StandardItemizeEnv=true [false]\MessageBreak
1585 StandardEnumerateEnv=true [false]\MessageBreak
1586 StandardItemLabels=true [false]\MessageBreak
1587 ItemLabels=\textendash, \textbullet,
1588 \protect\ding{43},... [\textendash]\MessageBreak
1589 ItemLabeli=\textendash, \textbullet,
1590 \protect\ding{43},... [\textendash]\MessageBreak
1591 ItemLabelii=\textendash, \textbullet,
1592 \protect\ding{43},... [\textendash]\MessageBreak
1593 ItemLabeliii=\textendash, \textbullet,
1594 \protect\ding{43},... [\textendash]\MessageBreak
1595 ItemLabeliv=\textendash, \textbullet,
1596 \protect\ding{43},... [\textendash]\MessageBreak
1597 FrenchFootnotes=false [true]\MessageBreak
1598 AutoSpaceFootnotes=false [true]\MessageBreak
1599 AutoSpacePunctuation=false [true]\MessageBreak
1600 OriginalTypewriter=true [false]\MessageBreak
1601 ThinColonSpace=true [false]\MessageBreak
1602 ThinSpaceInFrenchNumbers=true [false]\MessageBreak
1603 FrenchSuperscripts=false [true]\MessageBreak
1604 LowercaseSuperscripts=false [true]\MessageBreak
1605 PartNameFull=false [true]\MessageBreak
1606 SuppressWarning=true [false]\MessageBreak
1607 CustomiseFigTabCaptions=false [true]\MessageBreak
1608 OldFigTabCaptions=true [false]\MessageBreak
1609 SmallCapsFigTabCaptions=false [true]\MessageBreak
1610 INGullSpace=true [false]\MessageBreak
1611 InnerGullSingle=true [false]\MessageBreak
1612 EveryParGull=open, close, none [open]\MessageBreak
1613 EveryLineGull=open, close, none
1614 [open in LuaTeX, none otherwise]\MessageBreak
1615 og= <left quote character>, fg= <right quote character>%
1616 \MessageBreak
1617 *****%
1618 \MessageBreak\protect\frenchbsetup{ShowOptions}}
1619 \fi
1620 }

```

At `\begin{document}`, we have to provide an `\xspace` command in case the `xspace` package is not loaded, do some setup for `hyperref`'s bookmarks, execute `\FBprocess@options`, switch `LuaTeX` punctuation on and issue some warnings if necessary.

```

1621 \AtBeginDocument{%
1622   \providecommand*\xspace{\relax}%

```

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

```

1623   \ifdefined\pdfstringdefDisableCommands

```

```

1624 \pdfstringdefDisableCommands{%
1625   \let\up\relax
1626   \let\fu\relax
1627   \let\degre\textdegree
1628   \let\degres\textdegree
1629   \def\ieme{e\xspace}%
1630   \def\iemes{es\xspace}%
1631   \def\ier{er\xspace}%
1632   \def\iers{ers\xspace}%
1633   \def\iere{re\xspace}%
1634   \def\ieres{res\xspace}%
1635   \def\FrenchEnumerate#1{#1\degre\space}%
1636   \def\FrenchPopularEnumerate#1{#1\degre)\space}%
1637   \def\No{N\degre\space}%
1638   \def\no{n\degre\space}%
1639   \def\Nos{N\degre\space}%
1640   \def\nos{n\degre\space}%
1641   \def\FB@og{\guillemotleft\space}%
1642   \def\FB@fg{\space\guillemotright}%
1643   \def\at{@}%
1644   \def\circonflexe{\string^}%
1645   \def\tild{\string~}%
1646   \let\bsc\textsc
1647 }%
1648 \fi

```

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

```

1649 \FBprocess@options

```

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

```

1650 \ifFB@luatex@punct
1651   \activate@luatexpunct
1652 \fi

```

Some warnings are issued when output font encodings are not properly set. With XeLaTeX or LuaLaTeX, `fontspec.sty` and `xunicode.sty` should be loaded unless T1 encoded fonts are used through `luainputenc`, in the latter case `\FB@og` and `\FB@fg` have to be redefined; with (pdf)LaTeX, a warning is issued when OT1 encoding is in use at the `\begin{document}`. Mind that `\encodingdefault` is defined as ‘long’, defining `\FBOTone` with `\newcommand*` would fail!

```

1653 \ifFBunicode
1654   \ifdefined\DeclareUTFcharacter
1655   \else
1656     \@ifpackageloaded{luainputenc}{}%
1657     {\PackageWarning{frenchb.lda}%
1658      {Add \protect\usepackage{fontspec} to the\MessageBreak
1659      preamble of your document,}%
1660     }%
1661   \fi
1662 \else
1663   \begingroup \newcommand{\FBOTone}{OT1}%

```

```

1664      \ifx\encodingdefault\FB0Tone
1665      \PackageWarning{frenchb.ldb}%
1666      {OT1 encoding should not be used for French.%
1667      \MessageBreak
1668      Add \protect\usepackage[T1]{fontenc} to the
1669      preamble\MessageBreak of your document,}%
1670      \fi
1671    \endgroup
1672  \fi
1673 }

```

2.11 French lists

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

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

```

1674 \let\listORI\list
1675 \let\endlistORI\endlist
1676 \def\FB@listVsettings{%
1677   \setlength{\itemsep}{0.4ex plus 0.2ex minus 0.2ex}%
1678   \setlength{\parsep}{0.4ex plus 0.2ex minus 0.2ex}%
1679   \setlength{\topsep}{0.8ex plus 0.4ex minus 0.4ex}%
1680   \setlength{\partopsep}{0.4ex plus 0.2ex minus 0.2ex}%

```

`\parskip` is of type 'skip', its mean value only (*not the glue*) should be subtracted from `\topsep` and added to `\partopsep`, so convert `\parskip` to a 'dimen' using `\@tempdima`.

```

1681   \@tempdima=\parskip
1682   \addtolength{\topsep}{-\@tempdima}%
1683   \addtolength{\partopsep}{\@tempdima}%
1684 }
1685 \def\listFB#1#2{\listORI{#1}{\FB@listVsettings #2}}
1686 \let\endlistFB\endlist

```

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

- The '•' is never used in French `itemize`-lists, an emdash '—' or an endash '–' is preferred for all levels. The item label to be used in French is stored in `\FrenchLabelItem`, it defaults to '—' and can be changed using `\frenchbsetup{}` (see section 2.10).

- Vertical spacing between items, before and after the list, should be *null* with *no glue* added;
- In French the labels of itemize-lists are vertically aligned as follows:

<p>Text starting at ‘parindent’</p> <p>← Leftmargin</p> <p>— first item...</p> <p>— first second level item</p> <p>— next one...</p> <p>— second item...</p>
--

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

```

1687 \newcommand*\FrenchLabelItem{\textendash}
1688 \newcommand*\Frlabelitemi{\FrenchLabelItem}
1689 \newcommand*\Frlabelitemii{\FrenchLabelItem}
1690 \newcommand*\Frlabelitemiii{\FrenchLabelItem}
1691 \newcommand*\Frlabelitemiv{\FrenchLabelItem}

```

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

```

1692 \newlength\listindentFB
1693 \setlength{\listindentFB}{-1pt}
1694 \newlength\descindentFB
1695 \setlength{\descindentFB}{-1pt}
1696 \newlength\labelwidthFB
1697 \setlength{\labelwidthFB}{-1pt}

```

\FB@listHsettings **\FB@listHsettings** holds the new horizontal settings chosen for French lists itemize
\leftmarginFB and enumerate starting with version 2.6a. They are based on the look requested in
French for itemize-lists.

```

1698 \newlength\leftmarginFB
1699 \def\FB@listHsettings{%
1700   \leftmarginFB\labelwidthFB
1701   \advance\leftmarginFB \labelsep
1702   \bbl@for\FB@dp {1, 2, 3, 4, 5, 6}%
1703     {\csname leftmargin\romannumeral\FB@dp\endcsname \leftmarginFB}%
1704   \advance\leftmarginFB \listindentFB
1705   \leftmargin\csname leftmargin\ifnum\@listdepth=\@ne i\else
1706                                     ii\fi\endcsname
1707 }

```

\itemizeFB New environment for French itemize-lists.

\FB@itemizesettings **\FB@itemizesettings** does two things: first suppress all vertical spaces including
glue when option **ReduceListSpacing** is set, then set horizontal indentations accord-
ing to **\FB@listHsettings** unless option **ListOldLayout** is **true** (compatibility with
lists up to v. 2.5k).

```

1708 \def\FB@itemizesettings{%
1709   \ifFBReduceListSpacing
1710     \setlength{\itemsep}{\z@}%
1711     \setlength{\parsep}{\z@}%
1712     \setlength{\topsep}{\z@}%
1713     \setlength{\partopsep}{\z@}%
1714     \@tempdima=\parskip
1715     \addtolength{\topsep}{-\@tempdima}%
1716     \addtolength{\partopsep}{-\@tempdima}%
1717   \fi
1718   \settowidth{\labelwidth}{\csname\@itemitem\endcsname}%
1719   \ifFBListOldLayout
1720     \setlength{\leftmargin}{\labelwidth}%
1721     \addtolength{\leftmargin}{\labelsep}%
1722     \addtolength{\leftmargin}{\parindent}%
1723   \else
1724     \FB@listHsettings
1725   \fi
1726 }

```

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

```

1727 \def\itemizeFB{%
1728   \ifnum \@itemdepth >\thr@@\toodeep\else
1729     \advance\@itemdepth\@ne
1730     \edef\@itemitem{labelitem\romannumeral\the\@itemdepth}%
1731     \expandafter
1732     \listORI
1733     \csname\@itemitem\endcsname
1734     \FB@itemizesettings
1735   \fi
1736 }
1737 \let\enditemizeFB\endlistORI

1738 \def\labelitemsFB{%
1739   \let\labelitemi\Frlabelitemi
1740   \let\labelitemii\Frlabelitemii
1741   \let\labelitemiii\Frlabelitemiii
1742   \let\labelitemiv\Frlabelitemiv
1743   \ifdim\labelwidthFB<\z@
1744     \settowidth{\labelwidthFB}{\FrenchLabelItem}%
1745   \fi
1746   \ifdim\listindentFB<\z@
1747     \ifdim\parindent=\z@
1748       \setlength{\listindentFB}{1.5em}%
1749     \else
1750       \setlength{\listindentFB}{\parindent}%
1751     \fi
1752   \fi
1753   \ifdim\descindentFB<\z@
1754     \setlength{\descindentFB}{\listindentFB}%
1755   \fi

```

1756 }

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

```
1757 \def\enumerateFB{%
1758   \ifnum \@enumdepth >\thr@@\toodeep\else
1759     \advance\@enumdepth\@ne
1760     \edef\@enumctr{enum\romannumeral\the\@enumdepth}%
1761     \expandafter
1762     \list
1763       \csname label\@enumctr\endcsname
1764       {\FB@listHsettings
1765         \usecounter\@enumctr\def\makelabel##1{\hss\llap{##1}}}%
1766   \fi
1767 }
1768 \let\endenumerateFB\endlistORI
```

\descriptionFB Same tuning for the description environment (see `classes.dtx` for the original definition). Customisable length `\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`.

```
1769 \def\descriptionFB{%
1770   \list{}\FB@listHsettings
1771     \labelwidth\z@
1772     \itemindent-\leftmargin
1773     \ifnum\@listdepth=1
1774       \ifdim\descindentFB=\z@
1775         \ifdim\listindentFB>\z@
1776           \leftmargini\listindentFB
1777           \leftmargin\leftmargini
1778           \itemindent-\leftmargin
1779         \fi
1780       \else
1781         \advance\itemindent by \descindentFB
1782       \fi
1783     \fi
1784     \let\makelabel\descriptionlabel}%
1785 }
1786 \let\enddescriptionFB\endlistORI
```

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

```
\bbl@frenchlistlayout1787 \def\update@frenchlists{%
\bbl@nonfrenchlistlayout1788   \ifFBReduceListSpacing \let\list\listFB \fi
1789   \ifFBStandardItemizeEnv
1790   \else \let\itemize\itemizeFB \fi
1791   \ifFBStandardItemLabels
1792   \else \labelitemsFB \fi
```

```

1793 \ifFBStandardEnumerateEnv
1794 \else \let\enumerate\enumerateFB \let\description\descriptionFB \fi
1795 }

```

In order to ensure compatibility with packages customising lists, the command `\update@frenchlists` should not be included in `\extrasfrench` yet, so we also define `\FB@ufl` as `\relax`, it will be redefined as `\update@frenchlists` in due time ‘AtBeginDocument’ by `\FBprocess@options`, see p. 55.

```

1796 \def\FB@ufl{\relax}
1797 \def\bbl@frenchlistlayout{%
1798   \ifFBGlobalLayoutFrench
1799   \else
1800     \babel@save\list          \babel@save\itemize
1801     \babel@save\enumerate     \babel@save\description
1802     \babel@save\labelitemi    \babel@save\labelitemii
1803     \babel@save\labelitemiii  \babel@save\labelitemiv
1804   \fi
1805   \FB@ufl
1806 }
1807 \def\bbl@nonfrenchlistlayout{%
1808   \ifFBGlobalLayoutFrench
1809     \update@frenchlists
1810   \fi
1811 }
1812 \FB@addto{extras}{\bbl@frenchlistlayout}
1813 \FB@addto{noextras}{\bbl@nonfrenchlistlayout}

```

2.12 French indentation of sections

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

```

1814 \def\bbl@frenchindent{%
1815   \ifFBGlobalLayoutFrench\else\babel@save\@afterindentfalse\fi
1816   \ifFBIndentFirst
1817     \let\@afterindentfalse\@afterindenttrue
1818     \@afterindenttrue
1819   \fi}
1820 \def\bbl@nonfrenchindent{%
1821   \ifFBGlobalLayoutFrench
1822     \ifFBIndentFirst
1823       \@afterindenttrue
1824     \fi
1825   \fi}
1826 \FB@addto{extras}{\bbl@frenchindent}
1827 \FB@addto{noextras}{\bbl@nonfrenchindent}

```


2.13 Formatting footnotes

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

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

We 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 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 `\ifBFAutoSpaceFootnotes`.

```
1828 \AtBeginDocument{\@ifpackageloaded{bigfoot}%
1829                     {\PackageInfo{frenchb.ldb}%
1830                      {bigfoot package in use.\MessageBreak
1831                       frenchb will NOT customise footnotes;\MessageBreak
1832                       reported}}%
1833                     {\let\@footnotemarkORI\@footnotemark
1834                      \def\@footnotemarkFB{\leavevmode\unskip\unkern
1835                                           \,\@footnotemarkORI}%
1836                      \ifBFAutoSpaceFootnotes
1837                      \let\@footnotemark\@footnotemarkFB
1838                      \fi}%
1839                     }
```

We then define `\@makefntextFB`, a variant of `\@makefntext` which is responsible for the layout of footnotes, to match the specifications of the French ‘Imprimerie Nationale’: footnotes will be indented by `\parindentFFN`, numbers (if any) typeset on the baseline (instead of superscripts), 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).

The value of `\parindentFFN` will be redefined at the `\begin{document}`, as the maximum of `\parindent` and 1.5em *unless* it has been set in the preamble (the weird value 10in is just for testing whether `\parindentFFN` has been set or not).

```
1840 \newdimen\parindentFFN
1841 \parindentFFN=10in
```

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

```
1842 \newcommand*{\dotFFN}{.}
1843 \newcommand*{\kernFFN}{\kern .5em}
1844 \newlength\FBfnindent
```

`\@makefntextFB`’s definition is now tuned according to the document’s class for better compatibility.

Koma-script classes provide `\deffootnote`, a handy command to customise the footnotes' layout (see English manual `scrguien.pdf`); it redefines `\@makefntext` and `\@@makefnmark`. First, save the original definitions.

```
1845 \ifFB@koma
1846   \let\@makefntextORI\@makefntext
1847   \let\@@makefnmarkORI\@@makefnmark
```

`\@makefntextFB` and `\@@makefnmarkFB` will be used when option `FrenchFootnotes` is `true`.

```
1848   \deffootnote[\FBfnindent]{0pt}{\parindentFFN}%
1849               {\thefootnotemark\dotFFN\kernFFN}
1850   \let\@makefntextFB\@makefntext
1851   \let\@@makefnmarkFB\@@makefnmark
```

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

```
1852   \deffootnote[\parindentFFN]{0pt}{\parindentFFN}%
1853               {\textsuperscript{\thefootnotemark}}
1854   \let\@makefntextTH\@makefntext
1855   \let\@@makefnmarkTH\@@makefnmark
```

Restore the original definitions.

```
1856   \let\@makefntext\@makefntextORI
1857   \let\@@makefnmark\@@makefnmarkORI
1858 \fi
```

Definitions for the memoir class:

```
1859 \@ifclassloaded{memoir}
```

(see original definition in `memman.pdf`)

```
1860   {\newcommand{\@makefntextFB}[1]{%
1861     \def\footscript##1{##1\dotFFN\kernFFN}%
1862     \setlength{\footmarkwidth}{\FBfnindent}%
1863     \setlength{\footmarksep}{-\footmarkwidth}%
1864     \setlength{\footparindent}{\parindentFFN}%
1865     \makefootmark #1}%
1866   }{}}
```

Definitions for the beamer class:

```
1867 \@ifclassloaded{beamer}
```

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

```
1868   {\def\@makefntextFB#1{%
1869     \def\insertfootnotetext{#1}%
1870     \def\insertfootnotemark{\insertfootnotemarkFB}%
1871     \usebeamertemplate***{footnote}}%
1872   \def\insertfootnotemarkFB{%
1873     \usebeamercolor[fg]{footnote mark}%
1874     \usebeamerfont*{footnote mark}%
1875     \@thefnmark\dotFFN\kernFFN}%
1876   }{}}
```

Now the default definition of `\@makefnmarkFB` 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)!

```

1877 \providecommand*\insertfootnotemarkFB{%
1878   \parindent=\parindentFFN
1879   \rule\z@\footnotesep
1880   \setbox\@tempboxa\hbox{\@thefnmark}%
1881   \ifdim\wd\@tempboxa>\z@
1882     \llap{\@thefnmark}\dotFFN\kernFFN
1883   \fi}
1884 \providecommand\@makefnmarkFB[1]{\insertfootnotemarkFB #1}

```

The rest of `\@makefnmark`’s customisation is done at the `\begin{document}`. We save the original definition of `\@makefnmark`, and then redefine `\@makefnmark` according to the value of flag `\ifFBFrenchFootnotes` (true or false). Koma-script classes require a special treatment.

```

1885 \AtBeginDocument{%
1886   \ifpackageloaded{bigfoot}{}%
1887   {\ifdim\parindentFFN<10in
1888     \else
1889       \parindentFFN=\parindent
1890       \ifdim\parindentFFN<1.5em \parindentFFN=1.5em \fi
1891     \fi
1892     \settowidth{\FBfnindent}{\dotFFN\kernFFN}%
1893     \addtolength{\FBfnindent}{\parindentFFN}%
1894     \let\@makefnmarkORI\@makefnmark
1895     \ifFB@koma

```

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

```

1896       \let\@makefnmarkORI\@makefnmark
1897       \long\def\@makefnmark#1{%
1898         \ifFBFrenchFootnotes
1899           \ifx\footnote\thanks
1900             \let\@makefnmark\@makefnmarkTH
1901             \@makefnmarkTH{#1}%
1902           \else
1903             \let\@makefnmark\@makefnmarkFB
1904             \@makefnmarkFB{#1}%
1905           \fi
1906         \else
1907           \let\@makefnmark\@makefnmarkORI
1908           \@makefnmarkORI{#1}%
1909         \fi}%
1910     \else

```

Special add-on for the memoir class: `\maketitle` redefines `\@makefnmark` as `\makethanksmark` which is customised as follows to match the other notes’ vertical alignment.

```

1911       \ifclassloaded{memoir}%
1912       {\ifFBFrenchFootnotes

```

```

1913         \setlength{\thanksmarkwidth}{\parindentFFN}%
1914         \setlength{\thanksmarksep}{-\thanksmarkwidth}%
1915     \fi
1916 }{}%

```

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

```

1917     \@ifclassloaded{beamer}%
1918     {\ifFBFrenchFootnotes
1919         \ifdim\parindentFFN=1.5em\else
1920             \FBWarning{frenchb}{%
1921                 \protect\parindentFFN\space is ineffective%
1922                 \MessageBreak within the beamer class.\MessageBreak
1923                 Reported}%
1924         \fi
1925     \fi
1926 }{}%

```

Definition of \@makefntext for all classes other than koma-script:

```

1927     \long\def\@makefntext#1{%
1928         \ifFBFrenchFootnotes
1929             \@makefntextFB{#1}%
1930         \else
1931             \@makefntextORI{#1}%
1932         \fi}%
1933     \fi
1934 }%
1935 }

```

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

```

1936 \newcommand*{\AddThinSpaceBeforeFootnotes}{\FBAutoSpaceFootnotestru}
1937 \newcommand*{\FrenchFootnotes}{\FBFrenchFootnotestru}
1938 \newcommand*{\StandardFootnotes}{\FBFrenchFootnotesfalse}

```

2.14 Clean up and exit

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

```

1939 \FBclean@on@exit
1940 \let\FB@llc\loadlocalcfg
1941 \let\loadlocalcfg@gobble
1942 \ldf@finish\CurrentOption
1943 \let\loadlocalcfg\FB@llc

```

3 Change History

v2.0		
\FBtextellipsis: Added special case for LY1 encoding, see bug report from Bruno Voisin (2004/05/18).	46	
\bsc: \hbox dropped, replaced by \kern0pt.	38	
\captionsfrench: 'Fig.' changed to 'Figure' and 'Tab.' to 'Table'. . . .	41	
\datefrench: 2 '\relax' added in \today's definition.	34	
\nombre: \nombre now requires numprint.sty.	40	
General: \parindentFFN not changed if already defined (required by JA for cah-gut.cls).	65	
Added warning for OT1 encoding.	58	
Footnotes are now printed by default 'à la française' for the whole document.	65	
New command \frenchbsetup added for global customisation. . .	47	
v2.0b		
General: Footnotes: Just do nothing (except warning) when the bigfoot package is loaded.	65	
v2.0c		
\frenchbsetup: Option ThinSpaceInFrenchNumbers added.	47	
General: There is no need to define here numprint's command \npstylefrench, it will be redefined 'AtBeginDocument' by \FBprocess@options.	41	
v2.0d		
\frenchbsetup: Options og and fg changed: limit the definition to French so that quote characters can be used in German.	47	
v2.0e		
\frenchbsetup: New option: StandardLists.	47	
v2.0f		
\frenchbsetup: StandardLayout option had no effect on lists. Test moved to \FBprocess@options. . . .	47	
Two typos corrected in option StandardLists: [false] → [true]		
		and StandardLayout → StandardLists. 47
v2.0g		
\frenchbsetup: Revert previous change to StandardLayout. This option must set the three flags \FBReduceListSpacingfalse, \FBCompactItemizefalse, and \FBStandardItemLabeltrue instead of \FBStandardItemtrue, so that later options can still change their value before executing \FBprocess@options. Same thing for option StandardLists. . .	47	
v2.1a		
\datefrench: \today changed (correction in 2.0 was wrong: \today was printed without spaces in toc).	34	
\frenchbsetup: New option: FrenchSuperscripts to define \up as \fup or as \textsuperscript. . .	47	
New option: LowercaseSuperscripts.	47	
General: Command \fup added to produce better superscripts than \textsuperscript.	35	
v2.1b		
\fup: Command \fup changed to use real superscripts from fourier v. 1.6.	35	
General: Disable some commands in bookmarks.	58	
v2.1c		
\degres: Provide a temporary definition (hyperref safe) of \degres in case it has to be expanded in the preamble (by beamer's \title command for instance).	38	
\up: Provide a temporary definition (hyperref safe) of \up in case it has to be expanded in the preamble (by beamer's \title command for instance).	35	
General: Added commands \Nos and \nos.	37	

v2.1d	(suggested by JA).	65
General: Argument of \ProvidesLanguage changed above from ‘french’ to ‘frenchb’ (otherwise \listfiles prints no date/version information). The real name of current language (french) as to be corrected before calling \LdfInit.	12	
Avoid warning “\end occurred when \ifx ... incomplete” with LaTeX-2.09.	12	
v2.2a		
\frenchbsetup: Default values of flags changed: default now means ‘StandardLayout’, they will be changed to ‘FrenchLayout’ AtEndOfPackage only if french is \bbbl@main@language.	47	
The global layout of the document is no longer changed when frenchb is not the last option of babel (\bbbl@main@language). Suggested by Ulrike Fischer. . . .	47	
When frenchb is babel’s last option, French becomes the document’s main language, so GlobalLayoutFrench applies. . . .	47	
\fup: \newif and \newdimen moved before \ifLaTeXe to avoid an error with plainTeX.	35	
v2.3a		
\NoAutoSpaceBeforeFDP: \NoAutoSpaceBeforeFDP and \AutoSpaceBeforeFDP now set the flag \ifFBAutoSpacePunctuation accordingly (LaTeX only).	29	
\fup: \lowercase changed to \MakeLowercase as the former doesn’t work for non ASCII characters in encodings like applemac, utf-8,	35	
General: In LaTeX, frenchb no longer adds spaces before ‘high punctuation’ characters in computer code. Suggested by Yannis Haralambous.	29	
v2.3b		
General: New commands \dotFFN and \kernFFN for more flexibility		
v2.3c		
\ttfamilyFB: Commands \ttfamily, \rmfamily and \sffamily have to be robust. Bug introduced in 2.3a, pointed out by Manuel Pégourié-Gonnard.	29	
v2.3d		
\bbbl@nonfrenchindent: Bug correction: previous versions of frenchb set the flag \if@afterindent to false outside French which is correct for English but wrong for some languages like Spanish. Pointed out by Juan José Torrens.	64	
v2.3e		
\NoAutoSpaceBeforeFDP: Execute \AutoSpaceBeforeFDP also in LaTeX to define \FDP@colonspace: needed for tex4ht, pointed out by MPG.	29	
v2.4a		
\CaptionSeparator: \PackageWarning changed to \FBWarning (in case \@makecaption has been customised). \FBWarning is defined as \PackageWarning by default but can be made silent using \frenchbsetup, (suggested by MPG).	43	
\frenchbsetup: New option SuppressWarning.	47	
\ifFBXeTeX: Added a new ‘if’ \FBunicode and some \lccode definitions to \extrasfrench and \noextrasfrench.	14	
General: \PackageWarning changed to \FBWarning (when bigfoot package in use).	65	
v2.4c		
\frenchbsetup: In \ttfamilyFB, also cancel automatic spaces inside French guillemets coded as characters (see \frenchbsetup). . .	52	
\ttfamilyFB: In \ttfamilyFB, also cancel automatic spaces inside French guillemets entered as characters (see \frenchbsetup). . .	30	

v2.4d	\up: Command \up defined with \providecommand instead of \newcommand as \up may be defined elsewhere (catalan.ldf). Bug pointed out by Felip Manyé i Ballester.	35	regarding the status of the French “apostrophe”.	14
			General: Moved the \newcount command outside \ifFB@xetex@punct ... \fi (it broke Plain formats).	24
v2.5a	\FBthinspace: Define \FBthinspace for those who want to customise the width of the space before ; and co.	16	v2.5e	General:
	\captionsfrench: \emph deleted in \seename and \alsoname to match what is done for the other languages. Suggested by Marc Baudoin.	41		\pdfstringdefDisableCommands should redefine \FB@og and \FB@fg instead of \og and \fg so that it works also when quotes are entered as characters. Reported by Sébastien Gouezel.
	\fg: \og and \fg do not print correctly in English when using XeTeX or LuaTeX, fixed by using \textquotedblleft and \textquotedblright defined above.	32	v2.5f	\FBtextellipsis: Unicode fonts also provide a ready made character for \textellipsis, let’s just use it (reported by Maxime Chupin, 2011/06/04).
	\textquotedblright: Change \guillemotleft and \guillemotright definitions for Unicode and provide definitions for \textquotedblleft and \textquotedblright. Insures correct printing of quotes by \og and \fg in French and outside. .	30		General: Changed definitions of \at, \circonflexe, \tild, \boi and \degre for Unicode based engines.
	General: New command \NoAutoSpacing, suggested by MPG.	30	v2.5g	\FB@xetex@punct@french: XeTeXcharclass(es) for French quotes will be set to \FB@guilo and \FB@guilf by options ‘og’ and ‘fg’ in \frenchbsetup. French quotes should behave as normal characters by default in XeLaTeX as in LaTeX.
	Punctuation is no longer made active with XeTeX-based engines.	15		\frenchbsetup: When \ifFB@xetex@punct is true, ‘og’ and ‘fg’ options now set XeTeXcharclasses of these characters to \FB@guilo and \FB@guilf. Otherwise French quotes behave as normal characters (their XeTeXcharclass is 0).
v2.5b	\frenchbsetup: Do not use the test \iflanguage{french} to check whether French is the main language or not, as it might be erroneously positive when English is the main language and no hyphenation patterns are available for French. In this case \l@french and \l@english are 0. Pointed out by Günter Milde. ...	48		General: Redefine \degre, \degres \at \circonflexe and \tild for bookmarks. Add \fup also.
v2.5d	\ifFBXeTeX: Added two new ‘if’ \FBXeTeX and \FBLuaTeX as XeTeX and behave differently		v2.5h	\degres: textcomp.sty has changed. The test about \M@TS1 is no longer relevant, let’s change it.
			v2.5i	\FB@xetex@punct@french:

General: If <code>\@makecaption</code> is undefined, no warning.	44	french.cfg will be loaded (if found) instead of frenchb.cfg. NO NEED for .cfg files in French anyway. . .	68
New class <code>\FB@guilnul</code> for characters U+00A0 (Unicode nobreakspace) and U+202F (Unicode nobreakthinspace), to prevent frenchb from adding spurious spaces inside quotes. . .	25	In Plain, provide a substitute for <code>\PackageWarning</code> and <code>\PackageInfo</code>	13
v3.0a		Merging of <code>\captionsfrenchb</code> , <code>\captionsfrançais</code> with <code>\captionsfrench</code> deleted in favor of new babel 3.9 syntax.	43
<code>\CaptionSeparator</code> : Remove <code>\CaptionSeparatorORI</code> , use <code>\babel@save</code> instead.	43	More informative, less TeXnical warning about <code>\@makecaption</code> . . .	45
<code>\FB@fg</code> : Added explicit <code>\FBguillskip</code> for LuaTeX.	31	New flag <code>\ifFB@luatex@punct</code> for ‘high punctuation’ management with LuaTeX engines.	15
Definitions of <code>\FB@og</code> and <code>\FB@fg</code> now depend on punctuation handling (LuaTeX / XeTeX / active). . .	31	New handling of ‘high punctuation’ through callbacks with LuaTeX engines.	17
<code>\FBprocess@options</code> : Changed option <code>ThinColonSpace</code> to make it work also with LuaTeX.	55	No warning about <code>\@makecaption</code> for SMF classes. No warning either with LuaTeX or XeTeX engines. . .	44
With koma-script and memoir class, customise <code>\captionformat</code> and <code>\captiondelim</code>	57	Options processing completely reorganised, now <code>\babel@save</code> and <code>\babel@savevariable</code> are usable for French.	47
<code>\FBthinskip</code> : LuaTeX requires dimensions: two new skips <code>\FBcolonskip</code> and <code>\FBthinskip</code> . . .	16	Support for options <code>frenchb</code> , <code>français</code> , <code>canadien</code> , <code>acadian</code> changed.	12
<code>\captionsfrench</code> : Take advantage of babel’s <code>\SetString</code> commands for <code>captionnames</code>	41	Test <code>\ifXeTeX</code> changed to <code>\ifFBunicode</code> and ‘ <code>xltxtra</code> ’ changed to ‘ <code>fontspec</code> ’.	59
<code>\datefrench</code> : Take advantage of babel’s <code>\SetString</code> commands for <code>\datefrench</code> . Doesn’t work with Plain (yet?).	34	v3.0b	
<code>\descriptionFB</code> : Added <code>\listindentFB</code> to <code>\itemindent</code> . Suggested by Denis Bitouzé. . .	63	General: <code>frenchb.lua</code> was not found by Lua function <code>dofile</code> (not <code>kpathsea</code> aware). Call function <code>kpse.find_file</code> first, as suggested by Paul Gaborit.	23
<code>\extrasfrench</code> : Take advantage of babel’s <code>\babel@savevariable</code> to handle apostrophe’s <code>\lccode</code> . . .	14	Require <code>luatexbase</code> with LaTeX in case <code>fontspec</code> has not been loaded before babel.	17
<code>\frenchbsetup</code> : New options <code>OldFigTabCaptions</code> and <code>CustomiseFigTabCaptions</code>	47	v3.0c	
General: <code>\LdfInit</code> checks <code>\datefrench</code> instead of <code>\captionsfrench</code> to avoid a conflict with <code>papertex.cls</code> which loads <code>datetime.sty</code>	12	<code>\FB@fg</code> : Changed <code>\FBguill@spacing</code> (internal) to <code>\FBguillspace</code> (public).	31
<code>\bbl@nonfrenchguillemets</code> deleted, use <code>\babel@save</code> instead.	32	<code>\datefrench</code> : <code>\SetString</code> still does not work for Plain with babel 3.9k. Need to define <code>\datefrench</code> . . .	34
		<code>\frenchbsetup</code> : Activate option <code>StandardLists</code> when beamer class is loaded.	48
		New option <code>INGuillSpace</code>	47

General: frenchb requires babel-3.9i.	13	frenchb.lua: Add a check for null fid in french_punctuation (Tikz \nullfont). Bug pointed out by Paul Gaborit.	20
Just load luatexbase.sty instead of luaotfload.sty with plain formats.	17		
No need to define \l@french as \lang@french, babel.def (3.9j) takes care for this.	12	v3.1c frenchb.lua: Previous bug fix for null glues (v3.0c) did not work properly. Fixed now (I hope). Pointed out by Jacques André. . .	20
frenchb.lua: Null glues should not trigger space insertion before high punctuation. Bug pointed out by Benoit Rivet for the 'lstlisting' environment of the listings package.	20	v3.1d General: New section: issue warnings if packages listings, numprint and natbib are loaded too early or too late vs babel.	46
v3.1a \frenchbsetup: Codes "13 and "14 added for French quotes in T1-encoding. Support for older versions of LuaTeX and XeTeX dropped.	52	v3.1e \frenchbsetup: Corrected typo: SmallCapsFigTabCaptions instead of SmallCapsFigTabCaptions. Pointed out by Céline Chevalier. .	47
New options InnerGuillSingle, EveryParGuill and EveryLineGuill to control \frquote.	47	v3.1f \FBprocess@options: Bug fix for the beamer class: figure and table captions are now consistent with frenchb's documentation. Pointed out by Denis Bitouzé.	57
General: fontspec is not required for T1 fonts used with the luainputenc.sty package.	59	Definition of \captionformat and \captiondelim changed when option CustomiseFigTabCaptions is set to false.	57
Misplaced \fi for plain formats. . .	17	\FBthinspace: \FBthinspace is no longer a kern but a skip (frenchb adds a nobreak penalty before it). 16	
New command \frquote for imbedded or long French quotations.	32	General: \FBCaption@Separator changed when option CustomiseFigTabCaptions is set to false.	45
frenchb.lua: Added flag addgl which must also be true when prev or next is not a char (i.e. kern0 in «\texttt{a}»).	21	v3.1g \captionsfrench: \partname's definition depends now on flag PartNameFull. No need to redefine it in \frenchbsetup.	41
Codes 0x13 and 0x14 added for French quotes in T1-encoding. . .	18	Bug fix for koma-scripts classes: a spurious dot was added by the \partformat command.	42
Look ahead when next is a kern (i.e. in «\texttt{a} »).	22	\frenchbsetup: PartNameFull now just sets the flag, nothing to add to \captionsfrench when false. 47	
v3.1b \captionsfrench: Change \scshape to customisable \FBfigtabshape for \figurename and \tablename.	41	General: Lua function french_punctuation is now inserted at the end of the "kerning" callback (no priority)	
\fprimo): Removed \lowercase from definitions of \FrenchEnumerate, ... \No and co: \up already does the conversion.	37		
\frenchbsetup: New option SmallCapsFigTabCaptions.	47		
\ieres: Removed \lowercase from definitions of \ieme and co: \up already does the conversion. . . .	37		

instead of "hpack_filter" and "pre_linebreak_filter".	23	trigger space insertion before high punctuation. Add a check on \lastkip.	25
Use Babel defined loops \bbl@for instead of \@for borrowed from file ltcntrl.dtx (\@for is undefined in Plain).	24	General: (pdfTeX shorthands) test on \lastskip changed from 0pt to 1sp for active punctuation for consistency with XeTeX and LuaTeX.	27
frenchb.lua: Flag addgl set to false for '«' at the end of an \hbox or a paragraph or when followed by a null glue (i.e. springs).	22	v3.1i \FB@luatex@punct@french: Use \babel@save to save and restore \shorthandon and \shorthandoff.	23
flag addgl set to false for '»' at the beginning of an \hbox or a paragraph or a tabular 'l' and 'c' columns.	21	\FB@xetex@punct@french: Save and restore \XeTeXinterchartokenstate, \shorthandon, \shorthandoff using \babel@savevariable and \babel@save, \XeTeXcharclass(es) using \FB@savevariable@loop.	25
Node HLIST added; node TEMP added for the first node of \hboxes.	19	General: Add a variant of \babel@savevariable to save \XeTeXcharclass(es) in a loop.	25
v3.1h General: french.cfg from e-french conflicts with frenchb. Do NOT load it (no need for .cfg files with frenchb anyway).	68	frenchb.lua: font.getfont(fid) possibly returns nil even for a positive fid (i.e. AMS lcircle1.pfb). Reported by François Legendre.	19
v3.1i \frquote: \luatexlocalleftbox changed to \localleftbox by new LaTeX release 2015/10/01.	33	v3.1m frenchb.lua: new_glue_scaled returns nil in case of invalid font table (i.e. lcircle1.pfb). In such cases frenchb leaves the node list unchanged.	19
General: \nombre command changed when numprint.sty is not loaded: only one warning, no error.	41	v3.2a \fg: \xspace moved from \FB@fg to \fg: \xspace messes up \frquote, pointed out by Sonia Labetoulle. As a side effect \xspace is now active in \fg in and outside French.	32
Compatibility code added due to changes in the 2015/10/01 LaTeX release, see ltnews23.tex.	17	General: beamer.cls requires a specific definition of \@makefntextFB (pointed out by DB). The same is true for memoir and koma-script classes (done).	65
Remove restriction about loading numprint.sty after babel.	46	v3.2b \NoAutoSpacing: \NoAutoSpacing made robust.	30
v3.1j \frquote: \PackageWarning is undefined in Plain, use \fb@warning instead.	33	\ifFB@xetex@punct: New counter \FB@nonchar needed for non	
\fr@quote completely rewritten: \leavevmode added and explicitly save/retore \everypar and \localleftbox instead of using a group in order to ensure compatibility with package wrapfig.	33		
General: Loading luatexbase.sty is no longer needed with LaTeX release 2015/10/01 or later.	17		
v3.1k \FB@xetex@punct@french: Thin glues (less than 1sp) should not			

characters: it's value will be 4095 for new engines and 255 for older ones.	16	<code>\FB@spacing@off</code> and <code>\FB@spacing@on</code>	30
General: Load <code>lualatex.tex</code> for plain LuaTeX to ensure <code>\newattribute</code> is defined.	17	General: New LuaTeX attribute <code>\FB@spacing</code>	17
Warning added when the subcaption package is loaded before <code>babel/frenchb</code>	45	Newif <code>\ifFB@spacing</code> and new commands <code>\FB@spacingon</code> , <code>\FB@spacingoff</code> to control space tuning in French.	17
<code>frenchb.lua</code> : <code>glue_spec</code> removed; starting with LuaTeX 0.95, glue specifications fit in glue.	19	Switch <code>\ifFB@spacing</code> added to the four French shorthands.	27
v3.2c		v3.2d	
<code>\FB@xetex@punct@french</code> : Switch <code>\ifFB@spacing</code> added to all <code>\XeTeXinterchartoks</code> commands.	25	<code>\FBthinspace</code> : Corrected typo in <code>\FBthinspace</code> : 1.66672pt changed to 1.6667pt.	16
<code>\FBthinspace</code> : Change .16667em to .5\fontdimen2\font to get in XeTeX and pdfTeX the same spacing as in LuaTeX.	16	<code>\descriptionFB</code> : Changed <code>\listindentFB</code> to <code>\descindentFB</code> which defaults to <code>\listindentFB</code> . <code>\leftmargini</code> reduced when <code>\descindentFB</code> is null.	63
<code>\NoAutoSpacing</code> : New definition based on <code>\FB@spacing@off</code> common to all engines.	30	v3.2e	
<code>\frenchbsetup</code> : Add a warning about options <code>og/fg</code> for old XeTeX or LuaTeX engines requiring active characters.	52	<code>\DecimalMathComma</code> : <code>\DecimalMathComma</code> didn't work with LuaTeX. Fixed now.	39
<code>\ttfamilyFB</code> : New definitions of <code>\ttfamilyFB</code> and <code>co</code> , common to all engines, based on		General: Add missing redefinitions for <code>\leftmarginiv</code> , <code>\leftmarginvi</code> . Suggested by J.F. Burnol.	61
		v3.2f	
		<code>\DecimalMathComma</code> : Fixed conflict with the <code>icomma</code> package.	39