

A Babel language definition file for French

frenchb.dtx v3.2b, 2016/04/18

Daniel Flipo
daniel.flipo@free.fr

Contents

1 The French language	2
1.1 Basic interface	2
1.2 Customisation	4
1.2.1 <code>\frenchbsetup</code>	4
1.2.2 Captions	8
1.3 Hyphenation checks	9
1.4 Changes	9
2 The code	11
2.1 Initial setup	11
2.2 Punctuation	14
2.2.1 Punctuation with LuaTeX	16
2.2.2 Punctuation with XeTeX	22
2.2.3 Punctuation with standard (pdf)TeX	25
2.2.4 Punctuation switches common to all engines	27
2.3 Commands for French quotation marks	28
2.4 Date in French	32
2.5 Extra utilities	33
2.6 Formatting numbers	37
2.7 Caption names	39
2.8 Dots...	43
2.9 More checks about packages' loading order	44
2.10 Setup options: keyval stuff	45
2.11 French lists	57
2.12 French indentation of sections	61
2.13 Formatting footnotes	62
2.14 Clean up and exit	65
3 Change History	67

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 2.0 and v3.2b are listed in subsection 1.4 p. 9.

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 v3.2b and was last revised on 2016/04/18.

² For each item, hooks are provided to reset standard L^AT_EX settings or to emulate the behavior of former versions of 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. 7.

`\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)” (N^o N^{os} n^o and n^{os}) are obtained via the commands `\No`, `\Nos`, `\no`, `\nos`.
6. Two commands are provided to typeset the symbol for “degré”: `\degre` prints the raw character and `\degres` should be used to typeset temperatures (e.g., “20~\degres C” with an 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 be an ordinary character *in French only* (no space added); as a counterpart, if `\DecimalMathComma` is active, an explicit space has to be added in lists and intervals: `$(0,\ 1)$`, `$(x,\ y)$`. `\StandardMathComma` switches back to the standard behaviour of the comma.
8. A command `\nombre` was provided in 1.x versions to easily format numbers in slices of three digits separated either by a comma in English or with a space in French; `\nombre` is now mapped to `\numprint` from `numprint.sty`, see `numprint.pdf` for more information.
9. 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 remainder of the list of offered options. As usual with `keyval`

syntax, boolean options (as `ShowOptions`) can be entered as `ShowOptions=true` or just `ShowOptions`, the `=true` part can be omitted.

The other options are listed below. Their default value is shown between braces, sometimes followed by a `*`. The `*` means that the default shown applies when `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 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, `AutoSpacePunctuation` is locally switched to `false`, no spurious space is added in that case, so the default behaviour of babel-french in that area should be fine in most circumstances.

Choosing `AutoSpacePunctuation=false` will ensure that a proper space will be added before ‘:;!?’ *if and only if* a (normal) space has been typed in. Those who are unsure about their typing in this area should stick to the default option and use the provided `\NoAutoSpacing` command

inside a group in case an unwanted space is added by 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’.

`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 $\text{\LaTeX} 2_{\epsilon}$ I suggest this:

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

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

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

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

Frequent mismatches:

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

1.4 Changes

What's new in version 3.2?

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: the lua code included in frenchb.lua v. 3.2b 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 on older installations too.

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; set `\listindentFB=0pt` to get the former layout.
- The last but not least change affects the (recent) LuaTeX-based engines, (this means version 0.76 as included in TL2013 and up): active characters are no longer used in French for 'high punctuation'⁴. 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.

⁴The 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).

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

```

129 \newif\ifFBAutoSpaceGuill \FBAutoSpaceGuilltrue

```

`\ifFB@active@punct`

```

130 \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).

```

131 \newif\ifFB@luatex@punct
132 \ifBFLuaTeX
133   \ifnum\luatexversion>94
134     \FB@luatex@puncttrue\FB@active@punctfalse
135   \else
136     \ifx\PackageWarning\@undefined
137       \fb@warning{Please upgrade LuaTeX to version 0.95 or above!\\%
138         frenchb will make high punctuation characters (;:!) active\\%
139         with LuaTeX < 0.95.}%
140     \else
141       \PackageWarning{frenchb.ldf}{Please upgrade LuaTeX
142         to version 0.95 or above!\MessageBreak
143         frenchb will make high punctuation characters\MessageBreak
144         (;:!) active with LuaTeX < 0.95;\MessageBreak reported}%
145     \fi
146   \fi
147 \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 was increased from 256 to 4096 in XeTeX v. 0.99996, the class for non characters is now 4095 instead of 255.

```

148 \newcount\FB@nonchar
149 \newif\ifFB@xetex@punct
150 \begingroup\expandafter\expandafter\expandafter\endgroup
151 \expandafter\ifx\csname XeTeXinterchartokenstate\endcsname\relax
152 \else
153   \FB@xetex@puncttrue\FB@active@punctfalse
154   \ifdim\the\XeTeXversion\XeTeXrevision pt<0.99996pt
155     \FB@nonchar=255 \relax
156   \else
157     \FB@nonchar=4095 \relax
158   \fi
159 \fi

```

\FBcolonspace According to the I.N. specifications, the ‘:’ requires an inter-word space before it, the **\FBthinspace** other three require just a `\thinspace`. We define `\FBcolonspace` as `\space` (inter-word space) and `\FBthinspace` as `\thinspace` (both are user customisable). LuaTeX **\FBcolonskip** requires skips instead of commands, so we define `\FBcolonskip` and `\FBthinskip` to hold the specifications (width/stretch/shrink) of `\space` and `\thinspace` for the `\lmr10` font; these parameters will be scaled for the current font by the `frenchb.lua` script (see how p. 17). `\FBcolonskip` and `\FBthinskip` are also user customisable.

```

160 \newcommand*{\FBcolonspace}{\space}
161 \newcommand*{\FBthinspace}{\hskip .16667em \relax}
162 \newskip\FBcolonskip
163 \FBcolonskip=3.33pt plus 1.665pt minus 1.11pt \relax
164 \newskip\FBthinskip

```

```
165 \FBthinskip=1.66672pt \relax
```

2.2.1 Punctuation with LuaTeX

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

The following `\directlua` call ensures compatibility with LaTeX releases prior to 2015/10/01: the `\localleftbox` primitive⁵ introduced by Omega was prefixed with “`luatex`”, it should no longer be, see `ltnews23.tex` for details.

```
166 \ifFB@luatex@punct
167   \directlua{tex.enableprimitives("", tex.extraprimtives("omega"))}
```

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

```
168   \begingroup\expandafter\expandafter\expandafter\endgroup
169   \expandafter\ifx\csname newluafunction\endcsname\relax
```

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

```
170   \input l\luatex.tex
171   \fi
172   \newattribute\FB@addDPspace   \FB@addDPspace=1 \relax
173   \newattribute\FB@addGUILspace \FB@addGUILspace=0 \relax
174   \ifLaTeXe
175     \PackageInfo{frenchb.ldf}{No need for active punctuation
176                           characters\MessageBreak with this version
177                           of LuaTeX!\MessageBreak reported}
178   \else
179     \fb@info{No need for active punctuation characters\\
180             with this version of LuaTeX!}
181   \fi
182 \fi
```

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

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

```
183 local FB_punct_thin =
184   {[string.byte("!")] = true,
185    [string.byte("?")] = true,
186    [string.byte(";")] = true}
187 local FB_punct_thick =
188   {[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 ‘`»`’.

```
189 local FB_punct_left =
```

⁵used by `\frquote`, see p. 31.


```

190  {[string.byte("!")] = true,
191   [string.byte("?")] = true,
192   [string.byte(";")] = true,
193   [string.byte(":")] = true,
194   [0x14]                = true,
195   [0xBB]                = true}
196 local FB_punct_right =
197   {[0x13]                = true,
198    [0xAB]                = true}

```

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

```

199 local FB_punct_null =
200   {[string.byte("!")] = true,
201    [string.byte("?")] = true,
202    [string.byte("[") = true,
203    [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.

```

204   [0xA0]                = true,
205   [0x202F]              = true}
206 local FB_guil_null =
207   {[0xA0]                = true,
208    [0x202F]              = true}

```

Local definitions for nodes:

```

209 local new_node      = node.new
210 local copy_node     = node.copy
211 local node_id       = node.id
212 local HLIST         = node_id("hlist")
213 local TEMP          = node_id("temp")
214 local KERN          = node_id("kern")
215 local GLUE          = node_id("glue")
216 local GLYPH         = node_id("glyph")
217 local PENALTY       = node_id("penalty")
218 local nobreak       = new_node(PENALTY)
219 nobreak.penalty     = 10000
220 local insert_node_before = node.insert_before
221 local insert_node_after  = node.insert_after
222 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...

```

223 local thin10 = tex.skip['FBthinskip']
224 local thinwd = thin10.width/65536/3.33
225 local thinst = thin10.stretch/65536/1.665
226 local thinsh = thin10.shrink/655.36/1.11
227 local coln10 = tex.skip['FBcolonskip']
228 local colnwd = coln10.width/65536/3.33
229 local colnst = coln10.stretch/65536/1.665

```

```

230 local colnsh = coln10.shrink/65536/1.11
231 local guil10 = tex.skip['FBguillskip']
232 local guilwd = guil10.width/65536/3.33
233 local guilst = guil10.stretch/65536/1.665
234 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):

```

235 local font_table = {}
236 local function new_glue_scaled (fid,width,stretch,shrink)
237   if fid > 0 then
238     local fp = font_table[fid]
239     if not fp then
240       local ft = font.getfont(fid)
241       if ft then
242         font_table[fid] = ft.parameters
243         fp = font_table[fid]
244       end
245     end
246     local gl = new_node(GLUE,0)
247     if fp then
248       gl.width = width * fp.space
249       gl.stretch = stretch * fp.space_stretch
250       gl.shrink = shrink * fp.space_shrink
251       return gl
252     else
253       return nil
254     end
255   else
256     return nil
257   end
258 end

```

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

```

259 local addDPspace = luatexbase.attributes['FB@addDPspace']
260 local addGUILspace = luatexbase.attributes['FB@addGUILspace']
261 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).

```

262 local function french_punctuation (head)
263   for item in node.traverse_id(GLYPH, head) do
264     local lang = item.lang
265     local char = item.char
266     local fid = item.font

```

```

267     local SIG = has_attribute(item, addGUILspace)
268     if lang == FR and FB_punct_left[char] and fid > 0 then
269         local prev = item.prev
270         local prev_id, prev_subtype, prev_char
271         if prev then
272             prev_id = prev.id
273             prev_subtype = prev.subtype
274             if prev_id == GLYPH then
275                 prev_char = prev.char
276             end
277         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.

```

278     local is_glue = prev_id == GLUE
279     local glue_wd
280     if is_glue then
281         glue_wd = prev.width
282     end
283     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 to *glue* in the node list), we remove the *glue* and add a nobreak penalty and the required *glue*. Otherwise (auto option), the penalty and the required *glue* are inserted if attribute \FB@addDPspace is set, unless one of these three condition 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{ }.

```

284     if FB_punct_thin[char] or FB_punct_thick[char] then
285         local SBDP = has_attribute(item, addDPspace)
286         local auto = SBDP and SBDP > 0
287         if auto then
288             if (prev_char and FB_punct_null[prev_char]) or
289                 (is_glue and glue_wd <= 1) or
290                 (prev_id == HLIST and prev_subtype == 3) or
291                 (prev_id == TEMP) then
292                 auto = false
293             end
294         end
295         local fbglue
296         if FB_punct_thick[char] then
297             fbglue = new_glue_scaled(fid, colnwd, colnst, colnsh)
298         else
299             fbglue = new_glue_scaled(fid, thinwd, thinst, thinsh)
300         end

```

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

```

301     if (realglue or auto) and fbglue then

```

```

302         if realglue then
303             head = remove_node(head,prev,true)
304         end
305         insert_node_before(head, item, copy_node(nobreak))
306         insert_node_before(head, item, copy_node(fbg glue))
307     end

```

Let's consider '»' now (the only remaining glyph of FB_punct_left class): we just have to remove any *glue* possibly preceeding '»', 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.

```

308     elseif SIG and SIG > 0 then
309         local addgl = (prev_char and not FB_guil_null[prev_char]) or
310             (not prev_char and
311                 prev_id ~= TEMP and
312                 not (prev_id == HLIST and prev_subtype == 3)
313             )

```

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

```

314         if is_glue and glue_wd <= 1 then
315             addgl = false
316         end
317         local fbg glue = new_glue_scaled(fid,guilwd,guilst,guilsh)
318         if addgl and fbg glue then
319             if is_glue then
320                 head = remove_node(head,prev,true)
321             end
322             insert_node_before(head, item, copy_node(nobreak))
323             insert_node_before(head, item, copy_node(fbg glue))
324         end
325     end
326 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*.

```

327     if lang == FR and FB_punct_right[char] and fid > 0
328         and SIG and SIG > 0 then
329         local next = item.next
330         local next_id, next_subtype, next_char, nextnext, kern_wd
331         if next then
332             next_id = next.id
333             next_subtype = next_subtype
334             if next_id == GLYPH then
335                 next_char = next.char

```

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

```

336         elseif next_id == KERN then
337             kern_wd = next.kern
338             if kern_wd == 0 then
339                 nextnext = next.next
340                 if nextnext then
341                     next = nextnext
342                     next_id = nextnext.id
343                     next_subtype = nextnext.subtype
344                     if next_id == GLYPH then
345                         next_char = nextnext.char
346                     end
347                 end
348             end
349         end
350     end
351     local is_glue = next_id == GLUE
352     if is_glue then
353         glue_wd = next.width
354     end
355     local addgl = (next_char and not FB_guil_null[next_char]) or
356                 (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).

```

357     if is_glue and glue_wd == 0 then
358         addgl = false
359     end
360     local fid = item.font
361     local fbglue = new_glue_scaled(fid,guilwd,guilst,guilsh)
362     if addgl and fbglue then
363         if is_glue then
364             head = remove_node(head,next,true)
365         end
366         insert_node_after(head, item, copy_node(fbglue))
367         insert_node_after(head, item, copy_node(nobreak))
368     end
369 end
370 end
371 return head
372 end
373 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.

```

374 \ifFB@luatex@punct
375   \newcommand*{\FB@luatex@punct@french}{%
376     \babel@save{\shorthandon}%

```

```

377 \babel@save{\shorthandoff}%
378 \def\shorthandoff##1{%
379   \ifx\PackageWarning\@undefined
380     \fb@warning{\noexpand\shorthandoff{;:!?} is helpless with
381       LuaTeX,\ use \noexpand\NoAutoSpacing
382       *inside a group* instead.}%
383   \else
384     \PackageWarning{frenchb.ldf}{\protect\shorthandoff{;:!?} is
385       helpless with LuaTeX,\MessageBreak use \protect\NoAutoSpacing
386       \space *inside a group* instead;\MessageBreak reported}%
387   \fi}%
388 \def\shorthandon##1{%
389 }
390 \FB@addto{extras}{\FB@luatex@punct@french}

```

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

```

391 \def\activate@luatexpunct{%
392   \directlua{%
393     FR = \the\l@french
394     local path = kpse.find_file("frenchb.lua", "lua")
395     if path then
396       local f = dofile(path)
397       luatexbase.add_to_callback("kerning",
398         f, "frenchb.french_punctuation")
399     else
400       texio.write_nl('')
401       texio.write_nl('*****')
402       texio.write_nl('Error: frenchb.lua not found.')
403       texio.write_nl('*****')
404       texio.write_nl('')
405     end
406   }%
407 }
408 \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.

```

409 \ifFB@xetex@punct
410   \ifLaTeXe
411     \PackageInfo{frenchb.ldb}{No need for active punctuation characters%
412               \MessageBreak with this version of XeTeX!%
413               \MessageBreak reported}
414   \else
415     \fb@info{No need for active punctuation characters\\
416             with this version of XeTeX!}
417   \fi

```

Six new character classes are defined for babel-french.

```

418 \newXeTeXintercharclass\FB@punctthick
419 \newXeTeXintercharclass\FB@punctthin
420 \newXeTeXintercharclass\FB@punctnul
421 \newXeTeXintercharclass\FB@guilo
422 \newXeTeXintercharclass\FB@guilf
423 \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.

```

424 \def\FBsavevariable@loop#1#2{\begingroup
425   \toks@\expandafter{\originalTeX #1}%
426   \edef\x{\endgroup
427     \def\noexpand\originalTeX{\the\toks@ #2=\the#1#2\relax}}%
428   \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
)]	{	}	,	-	.	"	%	'	'	'

```

429 \def\FB@charlist{"21,"3A,"3B,"3F,"AB,"BB,"28,"5B,"A0,"202F,%
430               "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.

```

431 \newcommand*{\FB@xetex@punct@french}{%
432   \babel@savevariable{\XeTeXinterchartokenstate}%
433   \babel@save{\shorthandon}%

```

```

434 \babel@save{\shorthandoff}%
435 \bbl@for\FB@char\FB@charlist
436   {\FBsavevariable@loop{\XeTeXcharclass}{\FB@char}}%
437 \def\shorthandoff##1{%
438   \ifx\PackageWarning\undefined
439     \fb@warning{\noexpand\shorthandoff{;:!?} is helpless with
440       XeTeX,\ \ use \noexpand\NoAutoSpacing
441       *inside a group* instead.}%
442   \else
443     \PackageWarning{frenchb.ldf}{\protect\shorthandoff{;:!?} is
444       helpless with XeTeX,\MessageBreak use \protect\NoAutoSpacing
445       \space *inside a group* instead;\MessageBreak reported}%
446   \fi}%
447 \def\shorthandon##1{%

```

Let's now set the classes and interactions between classes.

```

448 \XeTeXinterchartokenstate=1
449 \XeTeXcharclass '\: = \FB@punctthick
450 \XeTeXinterchartoks \z@ \FB@punctthick = {%
451   \ifhmode\FDP@colonspace\fi}%
452 \XeTeXinterchartoks \FB@guilf \FB@punctthick = {%
453   \FDP@colonspace}%

```

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.

```

454 \XeTeXinterchartoks \FB@nonchar \FB@punctthick = {%
455   \ifhmode
456     \ifdim\lastskip>1sp
457       \unskip\penalty\@M\FBcolonspace
458     \else
459       \FDP@colonspace
460     \fi
461   \fi}%
462 \bbl@for\FB@char
463   {'\;, '\!, '\?}%
464   {\XeTeXcharclass\FB@char=\FB@punctthin}%
465 \XeTeXinterchartoks \z@ \FB@punctthin = {%
466   \ifhmode\FDP@thinspace\fi}%
467 \XeTeXinterchartoks \FB@guilf \FB@punctthin = {%
468   \FDP@thinspace}%
469 \XeTeXinterchartoks \FB@nonchar \FB@punctthin = {%
470   \ifhmode
471     \ifdim\lastskip>1sp
472       \unskip\penalty\@M\FBthinspace
473     \else
474       \FDP@thinspace
475     \fi
476   \fi}%

```



```

477 \XeTeXinterchartoks \FB@guilo \z@ = {%
478     \ifFBAutoSpaceGuill\FBguillspace\fi}%
479 \XeTeXinterchartoks \FB@guilo \FB@nonchar = {%
480     \ifFBAutoSpaceGuill\FBguillspace\ignorespaces\fi}%
481 \XeTeXinterchartoks \z@ \FB@guilf = {%
482     \ifFBAutoSpaceGuill\FBguillspace\fi}%
483 \XeTeXinterchartoks \FB@punctthin \FB@guilf = {%
484     \ifFBAutoSpaceGuill\FBguillspace\fi}%
485 \XeTeXinterchartoks \FB@nonchar \FB@guilf = {%
486     \ifFBAutoSpaceGuill\unskip\FBguillspace\fi}%

```

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

```

487 \bbl@for\FB@char
488     {\[, '\[, "A0, "202F}%
489     {\XeTeXcharclass\FB@char=\FB@punctnul}%

```

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

```

490 \bbl@for\FB@char
491     {\[, '\[, '\., '\-, '\), '\], '\}, '\%, "22, "27, "60, "2019}%
492     {\XeTeXcharclass\FB@char=\z@}%
493 }
494 \FB@addto{extras}{\FB@xetex@punct@french}

```

End of specific code for punctuation with modern XeTeX engines.

```
495 \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.

```

496 \ifFB@active@punct
497 \initiate@active@char{:}%
498 \initiate@active@char{;}%
499 \initiate@active@char{!}%
500 \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`.

```

501 \declare@shorthand{french}{;}{;%
502     \ifhmode
503         \ifdim\lastskip>1sp
504             \unskip\penalty\M\FBthinspace
505         \else
506             \FDP@thinspace
507         \fi

```

```
508      \fi
```

Now we can insert a ; character.

```
509      \string;}
```

The next three definitions are very similar.

```
510 \declare@shorthand{french}{!}{%
511     \ifhmode
512         \ifdim\lastskip>1sp
513             \unskip\penalty\@M\FBthinspace
514         \else
515             \FDP@thinspace
516         \fi
517     \fi
518     \string!}
519 \declare@shorthand{french}{?}{%
520     \ifhmode
521         \ifdim\lastskip>1sp
522             \unskip\penalty\@M\FBthinspace
523         \else
524             \FDP@thinspace
525         \fi
526     \fi
527     \string?}
528 \declare@shorthand{french}{:}{%
529     \ifhmode
530         \ifdim\lastskip>1sp
531             \unskip\penalty\@M\FBcolonspace
532         \else
533             \FDP@colonspace
534         \fi
535     \fi
536     \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.

```
537 \declare@shorthand{system}{:}{\string:}
538 \declare@shorthand{system}{!}{\string!}
539 \declare@shorthand{system}{?}{\string?}
540 \declare@shorthand{system}{;}{\string;}
541 %}
```

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

```
542 \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.

```
543     \bbl@activate{:}\bbl@activate{;}%
544     \bbl@activate{!}\bbl@activate{?}%
545 }
```

```

546 \FB@addto{noextras}{%
547   \bbl@deactivate{;}\bbl@deactivate{;}%
548   \bbl@deactivate{!}\bbl@deactivate{?}%
549 }
550 \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.

```

551 \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).

```

552 \def\autospace@beforeFDP{%
553   \ifFB@luatex@punct\FB@addDPspace=1 \fi
554   \def\FDP@thinspace{\penalty\@M\FBthinspace}%
555   \def\FDP@colonspace{\penalty\@M\FBcolonspace}}
556 \def\noautospace@beforeFDP{%
557   \ifFB@luatex@punct\FB@addDPspace=0 \fi
558   \let\FDP@thinspace\@empty
559   \let\FDP@colonspace\@empty}
560 \ifLaTeXe
561   \def\AutoSpaceBeforeFDP{\autospace@beforeFDP
562     \FBAutoSpacePunctuationtrue}
563   \def\NoAutoSpaceBeforeFDP{\noautospace@beforeFDP
564     \FBAutoSpacePunctuationfalse}
565   \AtEndOfPackage{\AutoSpaceBeforeFDP}
566 \else
567   \let\AutoSpaceBeforeFDP\autospace@beforeFDP
568   \let\NoAutoSpaceBeforeFDP\noautospace@beforeFDP
569   \AutoSpaceBeforeFDP
570 \fi

```

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

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

To be consistent with what is done for the ; : ! ? characters, `\ttfamilyFB` also switches off insertion of spaces inside French guillemets *when they are typed in as*

characters with the ‘og’/‘fg’ options in `\frenchbsetup{}`. This is also a workaround for the weird behaviour of these characters in verbatim mode.

```

571 \ifLaTeXe
572   \DeclareRobustCommand\ttfamilyFB{%
573     \FBAutoSpaceGuillfalse
574     \ifFB@luatex@punct\FB@addGUILspace=0 \fi
575     \noautospace@beforeFDP\ttfamilyORI}%
576   \DeclareRobustCommand\rmfamilyFB{%
577     \FBAutoSpaceGuilltrue
578     \ifFB@luatex@punct\FB@addGUILspace=1 \fi
579     \ifFBAutoSpacePunctuation
580       \autospace@beforeFDP
581     \else
582       \noautospace@beforeFDP
583     \fi
584     \rmfamilyORI}%
585   \DeclareRobustCommand\sffamilyFB{%
586     \FBAutoSpaceGuilltrue
587     \ifFB@luatex@punct\FB@addGUILspace=1 \fi
588     \ifFBAutoSpacePunctuation
589       \autospace@beforeFDP
590     \else
591       \noautospace@beforeFDP
592     \fi
593     \sffamilyORI}%
594 \fi

```

\NoAutoSpacing The following command will disable 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.

In case `\NoAutoSpacing` is used in moving arguments (footnotes, headers, etc.) `\noautospace@beforeFDP` will do the job (catcode changes are not taken into account, `\shorthandoff{;:!?}` does nothing).

```

595 \DeclareRobustCommand*\NoAutoSpacing{\FBAutoSpaceGuillfalse
596   \ifFB@active@punct\noautospace@beforeFDP\shorthandoff{;:!?}\fi
597   \ifFB@xetex@punct\XeTeXinterchartokenstate=0 \fi
598   \ifFB@luatex@punct\FB@addDPspace=0 \FB@addGUILspace=0 \fi
599 }

```

2.3 Commands for French quotation marks

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

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

```

600 \ifLaTeXe
601 \else
602   \ifFBunicode
603     \def\guillemotleft{{\char"00AB}}
604     \def\guillemotright{{\char"00BB}}
605     \def\textquotedblleft{{\char"201C}}
606     \def\textquotedblright{{\char"201D}}
607   \else
608     \def\guillemotleft{\leavevmode\raise0.25ex
609       \hbox{$\scriptscriptstyle\ll$}}
610     \def\guillemotright{\raise0.25ex
611       \hbox{$\scriptscriptstyle\gg$}}
612     \def\textquotedblleft{‘}
613     \def\textquotedblright{’}
614   \fi
615   \let\xspace\relax
616 \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.

```

617 \newskip\FBguillskip
618 \FBguillskip=2.664pt plus 0.500pt minus 0.888pt \relax
619 \newcommand*{\FBguillspace}{\penalty\@M\hskip.8\fontdimen2\font
620                               plus.3\fontdimen3\font
621                               minus.8\fontdimen4\font}

```

`\FBguillspace` is not used with LuaTeX.

```

622 \ifFB@luatex@punct
623   \DeclareRobustCommand*{\FB@og}{\leavevmode
624     \bgroup\FB@addGUILLspace=1 \guillemotleft\egroup}
625   \DeclareRobustCommand*{\FB@fg}{\ifdim\lastskip>\z@\unskip\fi
626     \bgroup\FB@addGUILLspace=1 \guillemotright\egroup}
627 \fi

```

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

```

628 \ifFB@xetex@punct
629   \DeclareRobustCommand*{\FB@og}{\leavevmode
630     \bgroup\FBAutoSpaceGuillfalse\guillemotleft\egroup
631     \FBguillspace}
632   \DeclareRobustCommand*{\FB@fg}{\ifdim\lastskip>\z@\unskip\fi
633     \FBguillspace

```

```

634      \bgroup\FBAutoSpaceGuillfalse\guillemotright\egroup}
635 \fi
636 \ifFB@active@punct
637   \DeclareRobustCommand*\FB@og{\leavevmode
638     \guillemotleft
639     \FBguillspace}
640   \DeclareRobustCommand*\FB@fg{\ifdim\lastskip>\z@ \unskip \fi
641     \FBguillspace
642     \guillemotright}
643 \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.

```

644 \newcommand*\og{\@empty}
645 \newcommand*\fg{\@empty}

```

The definitions of **\og** and **\fg** for quotation marks are switched on and off through the **\extrafrench \noextrafrench** 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).

```

646 \ifLaTeXe
647   \def\bbf@frenchguillemets{\renewcommand*\og{\FB@og}%
648     \renewcommand*\fg{\FB@fg\xspace}}
649   \renewcommand*\og{\textquotedblleft}
650   \renewcommand*\fg{\ifdim\lastskip>\z@ \unskip \fi
651     \textquotedblright\xspace}
652 \else
653   \def\bbf@frenchguillemets{\let\og\FB@og
654     \let\fg\FB@fg}
655   \def\og{\textquotedblleft}
656   \def\fg{\ifdim\lastskip>\z@ \unskip \fi \textquotedblright}
657 \fi

```

```

658 \FB@addto{extras}{\babel@save\og \babel@save\fg \bbf@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...

```

659 \newcommand*\ogi{\FB@og}
660 \newcommand*\fgi{\FB@fg}
661 \newcommand*\ogii{\textquotedblleft}
662 \newcommand*\fgii{\textquotedblright}

```

and the needed technical stuff to handle options:

```

663 \newcount\FBguill@level
664 \newtoks\FB@everypar
665 \newif\ifFBcloseguill \FBcloseguilltrue

```

```

666 \newif\ifFBInnerGuillSingle
667 \def\FBguillopen{\bgroup\NoAutoSpacing\guillemotleft\egroup}
668 \def\FBguillclose{\bgroup\NoAutoSpacing\guillemotright\egroup}
669 \let\FBguillnone\relax
670 \let\FBeveryparguill\FBguillopen
671 \ifFB@luatex@punct
672   \let\FBverylineguill\FBguillopen
673 \else
674   \let\FBverylineguill\FBguillnone
675 \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.

```

676 \ifLaTeXe
677   \DeclareRobustCommand\frquote{%
678     \@ifstar{\FBcloseguillfalse\fr@quote}%
679             {\FBcloseguilltrue\fr@quote}}
680 \else
681   \newcommand\frquote[1]{\fr@quote{#1}}
682 \fi

```

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

```

683 \newcommand{\fr@quote}[1]{%
684   \leavevmode
685   \advance\FBguill@level by \@ne

```

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

```

686 \def\FB@quotespace{\kern.8\fontdimen2\font}%
687 \ifcase\FBguill@level
688 \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:

```

689   \FB@everypar=\everypar
690   \ifx\FBeveryparguill\relax
691   \else
692     \def\FBeverypar@quote{\FBeveryparguill\FB@quotespace}%
693     \everypar=\expandafter{\the\everypar \FBeverypar@quote}%
694   \fi
695   \ogi #1\fgi
696 \or

```

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

```

697   \ifx\FBverylineguill\FBguillopen
698     \localleftbox{\guillemotleft\FB@quotespace}%
699     \let\FBeverypar@quote\relax
700     \ogi #1\ifFBcloseguill\fgi\fi

```

```

701 \else
702 \ifx\FBverylineguill\FBguillclose
703 \localleftbox{\guillemotright\FB@quotespace}%
704 \let\FBverypar@quote\relax
705 \ogi #1\ifFBcloseguill\fgi\fi
706 \else

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

707 \let\FBverypar@quote\relax
708 \ifFBInnerGuillSingle
709 \def\ogii{\leavevmode
710 \guilsinglleft\FBguillspace}%
711 \def\fgii{\ifdim\lastskip>\z@ \unskip\fi
712 \FBguillspace\guilsinglright}%
713 \ifx\FBveryparguill\FBguillopen
714 \def\FBverypar@quote{\guilsinglleft\FB@quotespace}%
715 \fi
716 \ifx\FBveryparguill\FBguillclose
717 \def\FBverypar@quote{\guilsinglright\FB@quotespace}%
718 \fi
719 \fi
720 \ogii #1\ifFBcloseguill \fgii \fi
721 \fi
722 \fi
723 \else

Warn if \FBguill@level  $\geq 3$ :

724 \ifx\PackageWarning\@undefined
725 \fb@warning{\noexpand\frquote\space accepts no more than
726 two levels.\\ Quotation not printed.}%
727 \else
728 \PackageWarning{frenchb.ldf}{%
729 \protect\frquote\space accepts no more than two levels
730 \MessageBreak Quotation not printed. Reported}
731 \fi
732 \fi

Clean on exit: adjust \FBguill@level and restore \localleftbox and \everypar.

733 \advance\FBguill@level by \m@ne
734 \ifx\FBverylineguill\FBguillnone\else\localleftbox{}\fi
735 \ifx\FBveryparguill\relax\else\everypar=\FB@everypar\fi
736 }

```

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.

```

737 \ifLaTeXe

```



```

738 \def\BabelLanguages{french,acadian}
739 \StartBabelCommands*\BabelLanguages}{date}
740     [unicode, fontenc=EU1 EU2, charset=utf8]
741     \SetString\monthiiname{février}
742     \SetString\monthviiiname{août}
743     \SetString\monthxiiname{décembre}
744 \StartBabelCommands*\BabelLanguages}{date}
745     \SetStringLoop{month#1name}{%
746         janvier,f\'evrier,mars,avril,mai,juin,juillet,%
747         ao\^ut,septembre,octobre,novembre,d\'ecembre}
748     \SetString\today{{\number\day}\ifnum1=\day {\ier}\fi\space
749         \csname month\romannumeral\month name\endcsname \space
750         \number\year
751     }
752 \EndBabelCommands
753 \else
754     \ifFBunicode
755         \@namedef{date\CurrentOption}{%
756             \def\today{{\number\day}\ifnum1=\day {\ier}\fi\space
757                 \ifcase\month
758                     \or janvier\or février\or mars\or avril\or mai\or
759                     juin\or juillet\or août\or septembre\or
760                     octobre\or novembre\or décembre\fi
761                 \space \number\year}}
762     \else
763         \@namedef{date\CurrentOption}{%
764             \def\today{{\number\day}\ifnum1=\day {\ier}\fi\space
765                 \ifcase\month
766                     \or janvier\or f\'evrier\or mars\or avril\or mai\or
767                     juin\or juillet\or ao\^ut\or septembre\or
768                     octobre\or novembre\or d\'ecembre\fi
769                 \space \number\year}}
770     \fi
771 \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 $\text{\LaTeX 2}_{\epsilon}$, 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).

```

772 \newif\ifFB@poorman
773 \newdimen\FB@Mht
774 \ifLaTeXe
775   \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{}`.

```

776 \newcommand*\FBsupR{-0.12}
777 \newcommand*\FBsupS{0.65}
778 \newcommand*\FB@lc[1]{\MakeLowercase{#1}}
779 \DeclareRobustCommand*\FB@up@fake[1]{%
780   \settoheight{\FB@Mht}{M}%
781   \addtolength{\FB@Mht}{\FBsupR \FB@Mht}%
782   \addtolength{\FB@Mht}{-\FBsupS ex}%
783   \raisebox{\FB@Mht}{\scalefont{\FBsupS}{\FB@lc{#1}}}%
784 }

```

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.

```

785 \def\FB@split#1#2#3#4\@nil{\def\FB@firstthree{#1#2#3}%
786                               \def\FB@suffix{#4}}
787 \def\FB@x{x}
788 \def\FB@j{j}
789 \DeclareRobustCommand*\FB@up[1]{%
790   \bgroup \FB@poormantrue
791   \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.

```

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

```

```

794     \reserved@a
795     {\ifx\FB@suffix\FB@x \FB@poormanfalse\fi
796     \ifx\FB@suffix\FB@j \FB@poormanfalse\fi
797     \ifFB@poorman \FB@up@fake{#1}%
798     \else          \FB@up@real{#1}%
799     \fi}%
800     {\FB@up@fake{#1}}%
801   \egroup}

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

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

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

804   \DeclareRobustCommand*\fup[1]{%
805     \ifx\realsuperscript\undefined
806       \FB@up{#1}%
807     \else
808       \bgroup\let\fakesuperscript\FB@up@fake
809         \realsuperscript{\FB@lc{#1}}\egroup
810     \fi}

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

811   \providecommand*\up{\relax}

Poor man's definition of \up for Plain.

812 \else
813   \providecommand*\up[1]{\leavevmode\raiselex\hbox{\sevenrm #1}}
814 \fi

```

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

```

\ier 815 \def\ieme{\up{e}\xspace}
\iere 816 \def\iemes{\up{es}\xspace}
\iemes 817 \def\ier{\up{er}\xspace}
\iers 818 \def\iers{\up{ers}\xspace}
\ieres 819 \def\iere{\up{re}\xspace}
      820 \def\ieres{\up{res}\xspace}

```

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

```

\no 821 \newcommand*\FrenchEnumerate[1]{%
\nos 822     #1\up{o}\kern+.3em}
\nos 823 \newcommand*\FrenchPopularEnumerate[1]{%
\primo 824     #1\up{o})\kern+.3em}

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

825 \def\primo{\FrenchEnumerate1}
826 \def\secundo{\FrenchEnumerate2}
827 \def\tertio{\FrenchEnumerate3}
828 \def\quarto{\FrenchEnumerate4}

```

while typing `\fprimo`) gives '1°).

```
829 \def\fprimo){\FrenchPopularEnumerate1}
830 \def\fsecundo){\FrenchPopularEnumerate2}
831 \def\ftertio){\FrenchPopularEnumerate3}
832 \def\fquarto){\FrenchPopularEnumerate4}
```

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

```
833 \DeclareRobustCommand*\No}{N\up{o}\kern+.2em}
834 \DeclareRobustCommand*\no}{n\up{o}\kern+.2em}
835 \DeclareRobustCommand*\Nos}{N\up{os}\kern+.2em}
836 \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}.

```
837 \DeclareRobustCommand*\bsc}[1]{\leavevmode\begin{group}\kern0pt
838                                     \scshape #1\end{group}}
839 \ifLaTeXe\else\let\scshape\relax\fi
```

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

```
840 \ifFBUnicode
841   \newcommand*\at{{\char"0040}}
842   \newcommand*\circonflexe{{\char"005E}}
843   \newcommand*\tild{{\char"007E}}
844   \newcommand*\boi{{\textbackslash}}
845   \newcommand*\degre{{\char"00B0}}
846 \else
847   \ifLaTeXe
848     \DeclareTextSymbol{\at}{T1}{64}
849     \DeclareTextSymbol{\circonflexe}{T1}{94}
850     \DeclareTextSymbol{\tild}{T1}{126}
851     \DeclareTextSymbolDefault{\at}{T1}
852     \DeclareTextSymbolDefault{\circonflexe}{T1}
853     \DeclareTextSymbolDefault{\tild}{T1}
854     \DeclareRobustCommand*\boi{{\textbackslash}}
855     \DeclareRobustCommand*\degre{\r{}}
856   \else
857     \def\T@one{T1}
858     \ifx\fontencoding\T@one
859       \newcommand*\degre{{\char6}}
860     \else
861       \newcommand*\degre{{\char23}}
862     \fi
863     \newcommand*\at{{\char64}}
```

```

864 \newcommand*{\circonflexe}{\char94}}
865 \newcommand*{\tild}{\char126}}
866 \newcommand*{\boi}{\backslash$}
867 \fi
868 \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.

```

869 \ifLaTeXe
870 \newcommand*{\degrees}{\degree}
871 \ifFBunicode
872 \DeclareRobustCommand*{\degrees}{\degree}
873 \else
874 \def\Warning@degree@TSone{%
875     \PackageWarning{frenchb.ldf}{%
876         Degrees would look better in TS1-encoding:%
877         \MessageBreak add \protect
878         \usepackage{textcomp} to the preamble.%
879         \MessageBreak Degrees used}}
880 \AtBeginDocument{\ifx\DeclareEncodingSubset\@undefined
881     \DeclareRobustCommand*{\degrees}{%
882         \leavevmode\hbox to 0.3em{\hss\degree\hss}%
883         \Warning@degree@TSone
884         \global\let\Warning@degree@TSone\relax}%
885     \else
886     \DeclareRobustCommand*{\degrees}{%
887         \hbox{\UseTextSymbol{TS1}{\textdegree}}}%
888     \fi
889 }
890 \fi
891 \else
892 \newcommand*{\degrees}{%
893     \leavevmode\hbox to 0.3em{\hss\degree\hss}}
894 \fi

```

2.6 Formatting numbers

\DecimalMathComma As mentioned in the T_EXbook p. 134, the comma is of type `\mathpunct` in math mode: it is automatically followed by a space. This is convenient in lists and intervals but unpleasant when the comma is used as a decimal separator in French: it has to be entered as `{,}`. `\DecimalMathComma` makes the comma be an ordinary character (of type `\mathord`) in French *only* (no space added); `\StandardMathComma` switches back to the standard behaviour of the comma.

```

895 \newcount\std@mcc
896 \newcount\dec@mcc
897 \std@mcc=\mathcode'\,
898 \dec@mcc=\std@mcc
899 \@tempcnta=\std@mcc
900 \divide\@tempcnta by "1000
901 \multiply\@tempcnta by "1000
902 \advance\dec@mcc by -\@tempcnta
903 \newcommand*\DecimalMathComma{\iflanguage{french}%
904     {\mathcode'\,=\dec@mcc}}}%
905 \FB@addto{extras}{\mathcode'\,=\dec@mcc}%
906 }
907 \newcommand*\StandardMathComma{\mathcode'\,=\std@mcc
908 \FB@addto{extras}{\mathcode'\,=\std@mcc}%
909 }
910 \FB@addto{noextras}{\mathcode'\,=\std@mcc}

```

\nombre The command `\nombre` is now borrowed from `numprint.sty` for $\text{\LaTeX} 2_{\epsilon}$. There is no point to maintain the former tricky code when a package is dedicated to do the same job and more. For Plain based formats, `\nombre` no longer formats numbers, it prints them as is and issues a warning about the change. Fake command `\nombre` for Plain based formats, warning users of `babel-french v. 1.x.` about the change:

```

911 \newcommand*\nombre[1]{\iflanguage{french}%
912     {\fb@warning{*** \noexpand\nombre
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.

```

913 \let\FBstop@here\relax
914 \def\FBclean@on@exit{\let\ifLaTeXe\undefined
915     \let\LaTeXettrue\undefined
916     \let\LaTeXefalse\undefined}
917 \ifx\magnification\@undefined
918 \else
919     \def\FBstop@here{\ifFB@luatexpunct
920         \activate@luatexpunct
921     \fi
922     \FBclean@on@exit
923     \ldf@quit\CurrentOption\endinput}
924 \fi
925 \FBstop@here

```

What follows is for $\text{\LaTeX} 2_{\epsilon}$ *only*; as all $\text{\LaTeX} 2_{\epsilon}$ based formats include $\text{\LaTeX} 2_{\epsilon}$, 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.

```

926 \renewcommand*\nombre[1]{\ifdefined\numprint%
927     {\fb@warning{*** \noexpand\nombre
no longer formats numbers\string! ***}}}%

```

```

928 \ifdefined\numprint
929     \numprint{#1}%
930 \else
931     \PackageWarning{frenchb.ldf}{%
932         \protect\nombre\space now relies on package numprint.sty,%
933         \MessageBreak add \protect
934         \usepackage[autolanguage]{numprint},\MessageBreak
935         see file numprint.pdf for more options.\MessageBreak
936         \protect\nombre\space called}%
937     \global\let\Warning@nombre\relax
938     {#1}%
939 \fi
940 }

```

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}`).

```

941 \ifx\FBfigtabshape\@undefined \let\FBfigtabshape\scshape \fi

```

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

```

942 \StartBabelCommands*{\BabelLanguages}{captions}
943     [unicode, fontenc=EU1 EU2, charset=utf8]
944     \SetString{\refname}{Références}
945     \SetString{\abstractname}{Résumé}
946     \SetString{\prefacename}{Préface}
947     \SetString{\contentsname}{Table des matières}
948     \SetString{\ccname}{Copie à }
949     \SetString{\proofname}{Démonstration}
950     \SetStringLoop{ordinal#1}{%
951         Première,Deuxième,Troisième,Quatrième,Cinquième,%
952         Sixième,Septième,Huitième,Neuvième,Dixième,Onzième,%
953         Douzième,Treizième,Quatorzième,Quinzième,Seizième,%
954         Dix-septième,Dix-huitième,Dix-neuvième,Vingtième}
955 \StartBabelCommands*{\BabelLanguages}{captions}
956     \SetString{\refname}{R\'ef\'erences}
957     \SetString{\abstractname}{R\'esum\'e}
958     \SetString{\bibname}{Bibliographie}
959     \SetString{\prefacename}{Pr\'eface}
960     \SetString{\chaptername}{Chapitre}
961     \SetString{\appendixname}{Annexe}
962     \SetString{\contentsname}{Table des mati\`eres}
963     \SetString{\listfigurename}{Table des figures}
964     \SetString{\listtablename}{Liste des tableaux}
965     \SetString{\indexname}{Index}

```

```

966 \SetString{\figurename}{\FBfigtabshape Figure}}
967 \SetString{\tablename}{\FBfigtabshape Table}}
968 \SetString{\pagename}{page}
969 \SetString{\seename}{voir}
970 \SetString{\alsoname}{voir aussi}
971 \SetString{\enclname}{P.~J. }
972 \SetString{\ccname}{Copie \a }
973 \SetString{\headtoname}{}
974 \SetString{\proofname}{D\’emonstration}
975 \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`.

```

976 \SetStringLoop{ordinal#1}{%
977     Premi\’ere,Deuxi\’eme,Troisi\’eme,Quatri\’eme,Cinqui\’eme,%
978     Sixi\’eme,Septi\’eme,Huiti\’eme,Neuvi\’eme,Dixi\’eme,Onzi\’eme,%
979     Douzi\’eme,Treizi\’eme,Quatorzi\’eme,Quinzi\’eme,Seizi\’eme,%
980     Dix-septi\’eme,Dix-huiti\’eme,Dix-neuvi\’eme,Vingti\’eme}
981 \AfterBabelCommands{%
982     \DeclareRobustCommand*\FB@emptypart{\def\thepart{}}%
983     \DeclareRobustCommand*\FB@partname{%
984         \ifFBPartNameFull
985             \csname ordinal\romannumeral\value{part}\endcsname\space
986             partie\FB@emptypart
987         \else
988             Partie%
989         \fi}%
990     }
991 \SetString{\partname}{\FB@partname}
992 \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. 42).

```

993 \AtEndOfPackage{%
994     \ifFB@koma
995         \ifdefined\partformat
996             \FB@addto{captions}{%
997                 \ifFBPartNameFull
998                     \babel@save\partformat
999                     \renewcommand*\partformat{\partname}%
1000                 \fi}%
1001         \fi
1002     \fi
1003 }

```

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 Lua \LaTeX and Xe \LaTeX , this glitch doesn't occur, you get 'Figure 1 :' which is correct in French. With pdf \LaTeX 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).

```

1004 \bgroup
1005 \catcode'::=12 \catcode'>:=12 \relax
1006 \long\gdef\STD@makecaption#1#2{%
1007   \vskip\abovcaptionskip
1008   \sbox\@tempboxa{#1: #2}%
1009   \ifdim \wd\@tempboxa >\hsize
1010     #1: #2\par
1011   \else
1012     \global \@minipagefalse
1013     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
1014   \fi
1015   \vskip\belowcaptionskip}
1016 \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.

```

1017 \newif\if@FBwarning@capsep
1018 \@FBwarning@capseptrue
1019 \newcommand{\FBWarning}[2]{\PackageWarning{#1}{#2}}
1020 \newcommand*\CaptionSeparator{\space\textendash\space}
1021 \def\FBCaption@Separator{: }
1022 \long\def\FB@makecaption#1#2{%
1023   \vskip\abovcaptionskip
1024   \sbox\@tempboxa{#1\FBCaption@Separator #2}%
1025   \ifdim \wd\@tempboxa >\hsize
1026     #1\FBCaption@Separator #2\par
1027   \else
1028     \global \@minipagefalse

```

```

1029 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
1030 \fi
1031 \vskip\belowcaptionskip}

```

Disable the standard warning with AMS and SMF classes.

```

1032 \@ifclassloaded{amsart}{\@FBwarning@capsepfalse}{}
1033 \@ifclassloaded{amsbook}{\@FBwarning@capsepfalse}{}
1034 \@ifclassloaded{amsdtx}{\@FBwarning@capsepfalse}{}
1035 \@ifclassloaded{amslatex}{\@FBwarning@capsepfalse}{}
1036 \@ifclassloaded{amproc}{\@FBwarning@capsepfalse}{}
1037 \@ifclassloaded{smfart}{\@FBwarning@capsepfalse}{}
1038 \@ifclassloaded{smfbook}{\@FBwarning@capsepfalse}{}

```

Disable the standard warning unless high punctuation is active.

```

1039 \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).

```

1040 \newif\ifFB@koma
1041 \@ifclassloaded{memoir}{\@FBwarning@capsepfalse}{}
1042 \@ifclassloaded{scrartcl}{\@FBwarning@capsepfalse\FB@komatrue}{}
1043 \@ifclassloaded{scrbook}{\@FBwarning@capsepfalse\FB@komatrue}{}
1044 \@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).

```

1045 \@ifclassloaded{beamer}{\@FBwarning@capsepfalse}{}
1046 \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.

```

1047 \@ifpackageloaded{caption}
1048   {\FBWarning{frenchb.ldb}%
1049     {Please load the "caption" package\MessageBreak
1050       AFTER babel/frenchb; reported}%
1051   \@FBwarning@capsepfalse}%
1052   {}

```

Same for package subcaption.

```

1053 \@ifpackageloaded{subcaption}
1054   {\FBWarning{frenchb.ldb}%
1055     {Please load the "subcaption" package\MessageBreak
1056       AFTER babel/frenchb; reported}%
1057   \@FBwarning@capsepfalse}%
1058   {}

```

Same for package floatrow.

```

1059 \@ifpackageloaded{floatrow}
1060   {\FBWarning{frenchb.ldb}%

```

```

1061      {Please load the "floatrow" package\MessageBreak
1062      AFTER babel/frenchb; reported}%
1063      \@FBwarning@capsepfalse}%
1064      {}

```

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’).

```

1065 \AtBeginDocument{%
1066   \ifx\@makecaption\STD@makecaption
1067     \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.

```

1068     \ifFBoldFigTabCaptions
1069     \else
1070       \def\FBCaption@Separator{{\autospace@beforeFDP : }}%
1071     \fi
1072     \ifFBCustomiseFigTabCaptions
1073       \ifx\bbbl@main@language\FB@french
1074         \def\FBCaption@Separator{\CaptionSeparator}%
1075       \fi
1076     \fi
1077     \@FBwarning@capsepfalse
1078   \fi
1079   \if@FBwarning@capsep
1080     \FBWarning{frenchb.ldb}%
1081     {Figures’ and tables’ captions might look like\MessageBreak
1082     ‘Figure 1:’ which is wrong in French.\MessageBreak
1083     Check your class or packages to change this;\MessageBreak
1084     reported}%
1085   \fi
1086   \let\FB@makecaption\relax
1087   \let\STD@makecaption\relax
1088 }

```

2.8 Dots...

\FBtextellipsis L^AT_EX 2_ε’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 L^AT_EX 2_ε only).

The \if construction in the L^AT_EX 2_ε definition of \dots doesn’t allow the use of xspace (xspace is always followed by a \fi), so we use the AMS-L^AT_EX 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.

```

1089 \ifFBunicode
1090   \let\FBtextellipsis\textellipsis
1091 \else
1092   \DeclareTextSymbol{\FBtextellipsis}{LY1}{133}
1093   \DeclareTextCommandDefault{\FBtextellipsis}{%
1094     .\kern\fontdimen3\font.\kern\fontdimen3\font.\xspace}
1095 \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 ‘At-BeginDocument’, if `amsmath` has not been loaded. `\Mdots@` doesn’t change when switching from/to French, while `\Tdots@` is redefined as `\FBtextellipsis` in French.

```

1096 \newcommand*{\Tdots@}{\@xp\textellipsis}
1097 \newcommand*{\Mdots@}{\@xp\mdots@}
1098 \AtBeginDocument{\DeclareRobustCommand*\dots}{\relax
1099   \csname\ifmmode M\else T\fi dots@endcsname}%
1100   \ifdefined\@xp\else\let\@xp\relax\fi
1101   \ifdefined\mdots@else\let\Mdots@\mathellipsis\fi
1102 }
1103 \def\bbl@frenchdots{\babel@save\Tdots@ \let\Tdots@\FBtextellipsis}
1104 \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).

```

1105 \ifFB@active@punct
1106   \@ifpackageloaded{listings}
1107     {\FBWarning{frenchb.ldf}%
1108       {Please load the "listings" package\MessageBreak
1109         AFTER babel/frenchb; reported}%
1110     }{}
1111 \fi

```

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

```

1112 \newif\if@FBwarning@natbib
1113 \ifFB@active@punct
1114   \@ifpackageloaded{natbib}{\if@FBwarning@natbibtrue}
1115 \fi
1116 \AtBeginDocument{%
1117   \if@FBwarning@natbib
1118     \@ifpackageloaded{natbib}{\if@FBwarning@natbibfalse}%
1119   \fi
1120   \if@FBwarning@natbib
1121     \FBWarning{frenchb.ldf}%
1122     {Please load the "natbib" package\MessageBreak
1123       BEFORE babel/frenchb; reported}%
1124   \fi
1125 }

```

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.

```
1126 \newcommand*{\frenchbsetup}[1]{%
1127   \setkeys{FB}{#1}%
1128 }%
1129 \@onlypreamble\frenchbsetup
```

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

```
1130 \newif\ifFBShowOptions           \FBShowOptionsfalse
1131 \newif\ifFBStandardLayout         \FBStandardLayouttrue
1132 \newif\ifFBGlobalLayoutFrench     \FBGlobalLayoutFrenchtrue
1133 \newif\ifFBReduceListSpacing      \FBReduceListSpacingfalse
1134 \newif\ifFBListOldLayout          \FBListOldLayoutfalse
1135 \newif\ifFBCompactItemize         \FBCompactItemizefalse
1136 \newif\ifFBStandardItemizeEnv     \FBStandardItemizeEnvtrue
1137 \newif\ifFBStandardEnumerateEnv   \FBStandardEnumerateEnvtrue
1138 \newif\ifFBStandardItemLabels     \FBStandardItemLabelstrue
1139 \newif\ifFBStandardLists          \FBStandardListstrue
1140 \newif\ifFBIndentFirst            \FBIndentFirstfalse
1141 \newif\ifFBFrenchFootnotes        \FBFrenchFootnotesfalse
1142 \newif\ifFBAutoSpaceFootnotes     \FBAutoSpaceFootnotesfalse
1143 \newif\ifFBOriginalTypewriter     \FBOriginalTypewriterfalse
1144 \newif\ifFBThinColonSpace         \FBThinColonSpacefalse
1145 \newif\ifFBThinSpaceInFrenchNumbers \FBThinSpaceInFrenchNumbersfalse
1146 \newif\ifFBFrenchSuperscripts     \FBFrenchSuperscriptstrue
1147 \newif\ifFBLowercaseSuperscripts  \FBLowercaseSuperscriptstrue
1148 \newif\ifFBPartNameFull           \FBPartNameFulltrue
1149 \newif\ifFBCustomiseFigTabCaptions \FBCustomiseFigTabCaptionsfalse
1150 \newif\ifFBOldFigTabCaptions     \FBOldFigTabCaptionsfalse
1151 \newif\ifFBSmallCapsFigTabCaptions \FBSmallCapsFigTabCaptionstrue
1152 \newif\ifFBSuppressWarning        \FBSuppressWarningfalse
1153 \newif\ifFBINGuillSpace           \FBINGuillSpacefalse
```

The defaults values of these flags have been chosen 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.

```

1154 \edef\FB@french{\CurrentOption}
1155 \AtEndOfPackage{%
1156   \ifx\bbl@main@language\FB@french
1157     \FBGlobalLayoutFrenchtrue
1158     \@ifclassloaded{beamer}%
1159       {\PackageInfo{frenchb.ldf}{%
1160         No list customisation for the beamer class,%
1161         \MessageBreak reported}}%
1162       {\FBReduceListSpacingtrue
1163        \FBStandardItemizeEnvfalse
1164        \FBStandardEnumerateEnvfalse
1165        \FBStandardItemLabelsfalse}%
1166     \FBIndentFirsttrue
1167     \FBFrenchFootnotesttrue
1168     \FBAutoSpaceFootnotesttrue
1169     \FBCustomiseFigTabCaptionstrue
1170   \else
1171     \FBGlobalLayoutFrenchfalse
1172   \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.

```

1173 \RequirePackage{keyval}%
1174 \define@key{FB}{ShowOptions}[true]%
1175   {\csname FBShowOptions#1\endcsname}%
1176 \define@key{FB}{StandardLayout}[true]%
1177   {\csname FBStandardLayout#1\endcsname
1178    \ifFBStandardLayout
1179      \FBReduceListSpacingfalse
1180      \FBStandardItemizeEnvtrue
1181      \FBStandardItemLabelstrue
1182      \FBStandardEnumerateEnvtrue
1183      \FBIndentFirstfalse
1184      \FBFrenchFootnotesfalse
1185      \FBAutoSpaceFootnotesfalse
1186      \FBGlobalLayoutFrenchfalse
1187    \else
1188      \FBReduceListSpacingtrue
1189      \FBStandardItemizeEnvfalse
1190      \FBStandardItemLabelsfalse
1191      \FBStandardEnumerateEnvfalse
1192      \FBIndentFirsttrue
1193      \FBFrenchFootnotesttrue

```

```

1194             \FBAutoSpaceFootnotesttrue
1195             \fi}%
1196 \define@key{FB}{GlobalLayoutFrench}[true]%
1197             {\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.

1198             \ifFBGlobalLayoutFrench
1199             \ifx\bbbl@main@language\FB@french
1200             \else
1201             \PackageWarning{frenchb.ldb}%
1202             {Option 'GlobalLayoutFrench' skipped:%
1203             \MessageBreak French is *not*
1204             babel's last option.\MessageBreak}%
1205             \fi
1206             \fi}%
1207 \define@key{FB}{ReduceListSpacing}[true]%
1208             {\csname FBReduceListSpacing#1\endcsname}%
1209 \define@key{FB}{ListOldLayout}[true]%
1210             {\csname FBListOldLayout#1\endcsname
1211             \ifFBListOldLayout
1212             \FBStandardEnumerateEnvtrue
1213             \renewcommand*\FrenchLabelItem{\textendash}%
1214             \fi}%
1215 \define@key{FB}{CompactItemize}[true]%
1216             {\csname FBCompactItemize#1\endcsname
1217             \ifFBCompactItemize
1218             \FBStandardItemizeEnvfalse
1219             \FBStandardEnumerateEnvfalse
1220             \else
1221             \FBStandardItemizeEnvtrue
1222             \FBStandardEnumerateEnvtrue
1223             \fi}%
1224 \define@key{FB}{StandardItemizeEnv}[true]%
1225             {\csname FBStandardItemizeEnv#1\endcsname}%
1226 \define@key{FB}{StandardEnumerateEnv}[true]%
1227             {\csname FBStandardEnumerateEnv#1\endcsname}%
1228 \define@key{FB}{StandardItemLabels}[true]%
1229             {\csname FBStandardItemLabels#1\endcsname}%
1230 \define@key{FB}{ItemLabels}{%
1231             \renewcommand*\FrenchLabelItem{#1}}%
1232 \define@key{FB}{ItemLabeli}{%
1233             \renewcommand*\Frlabelitemi{#1}}%
1234 \define@key{FB}{ItemLabelii}{%
1235             \renewcommand*\Frlabelitemii{#1}}%
1236 \define@key{FB}{ItemLabeliii}{%
1237             \renewcommand*\Frlabelitemiii{#1}}%
1238 \define@key{FB}{ItemLabeliv}{%
1239             \renewcommand*\Frlabelitemiv{#1}}%
1240 \define@key{FB}{StandardLists}[true]%

```

```

1241         {\csname FBStandardLists#1\endcsname
1242         \ifFBStandardLists
1243             \FBReduceListSpacingfalse
1244             \FBCompactItemizefalse
1245             \FBStandardItemizeEnvtrue
1246             \FBStandardEnumerateEnvtrue
1247             \FBStandardItemLabelstrue
1248         \else
1249             \FBReduceListSpacingtrue
1250             \FBCompactItemizetrue
1251             \FBStandardItemizeEnvfalse
1252             \FBStandardEnumerateEnvfalse
1253             \FBStandardItemLabelsfalse
1254         \fi}%
1255 \define@key{FB}{IndentFirst}[true]%
1256         {\csname FBIndentFirst#1\endcsname}%
1257 \define@key{FB}{FrenchFootnotes}[true]%
1258         {\csname FBFrenchFootnotes#1\endcsname}%
1259 \define@key{FB}{AutoSpaceFootnotes}[true]%
1260         {\csname FBAutoSpaceFootnotes#1\endcsname}%
1261 \define@key{FB}{AutoSpacePunctuation}[true]%
1262         {\csname FBAutoSpacePunctuation#1\endcsname}%
1263 \define@key{FB}{OriginalTypewriter}[true]%
1264         {\csname FBOriginalTypewriter#1\endcsname}%
1265 \define@key{FB}{ThinColonSpace}[true]%
1266         {\csname FBThinColonSpace#1\endcsname}%
1267 \define@key{FB}{ThinSpaceInFrenchNumbers}[true]%
1268         {\csname FBThinSpaceInFrenchNumbers#1\endcsname}%
1269 \define@key{FB}{FrenchSuperscripts}[true]%
1270         {\csname FBFrenchSuperscripts#1\endcsname}%
1271 \define@key{FB}{LowercaseSuperscripts}[true]%
1272         {\csname FBLowercaseSuperscripts#1\endcsname}%
1273 \define@key{FB}{PartNameFull}[true]%
1274         {\csname FBPartNameFull#1\endcsname}%
1275 \define@key{FB}{CustomiseFigTabCaptions}[true]%
1276         {\csname FBCustomiseFigTabCaptions#1\endcsname}%
1277 \define@key{FB}{OldFigTabCaptions}[true]%
1278         {\csname FBOldFigTabCaptions#1\endcsname}
    \CurrentOption no longer defined. It's value has been saved in \FB@CurOpt while
    reading frenchb.ldf.
1279         \ifFBOldFigTabCaptions
1280             \FB@addto{extras}{\babel@save\FBCaption@Separator
1281                 \def\FBCaption@Separator{\CaptionSeparator}}%
1282         \fi}%
1283 \define@key{FB}{SmallCapsFigTabCaptions}[true]%
1284         {\csname FBSmallCapsFigTabCaptions#1\endcsname
1285         \ifFBSmallCapsFigTabCaptions
1286             \let\FBfigtabshape\scshape
1287         \else
1288             \let\FBfigtabshape\relax

```



```

1289         \fi}%
1290 \define@key{FB}{SuppressWarning}[true]%
1291         {\csname FBSuppressWarning#1\endcsname
1292         \ifBSuppressWarning
1293         \renewcommand{\FBWarning}[2]{\relax}%
1294         \fi}%

```

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

```

1295 \define@key{FB}{INGuillSpace}[true]%
1296         {\csname FBINGuillSpace#1\endcsname}%
1297 \define@key{FB}{InnerGuillSingle}[true]%
1298         {\csname FBInnerGuillSingle#1\endcsname}%
1299 \define@key{FB}{EveryParGuill}{\expandafter\let\expandafter
1300         \FBeveryparguill\csname FBguill#1\endcsname}%
1301 \define@key{FB}{EveryLineGuill}{\expandafter\let\expandafter
1302         \FBeverylineguill\csname FBguill#1\endcsname
1303         \ifFB@luatex@punct
1304         \else
1305         \let\FBeverylineguill\FBguillnone
1306         \PackageWarning{frenchb.ldf}%
1307             {Option 'EveryLineGuill' skipped:%
1308             \MessageBreak this option is for
1309             LuaTeX *only*.\MessageBreak Reported}%
1310         \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@addGUILLspace` attribute for LuaTeX or set `\XeTeXcharclass` of quotes to the proper value for XeTeX.

```

1311 \define@key{FB}{og}{%
1312     \ifFBunicode

```

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

```

1313         \ifFB@luatex@punct
1314         \FB@addGUILLspace=1 \relax
1315         \fi

```

then with XeTeX it is a bit more tricky:

```

1316         \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).

```

1317         \XeTeXcharclass"13 = \FB@guilo
1318         \XeTeXcharclass"AB = \FB@guilo
1319         \XeTeXcharclass"A0 = \FB@guilnul
1320         \XeTeXcharclass"202F = \FB@guilnul
1321     \fi
1322 \else

```

This is for conventional TeX engines:

```

1323     \newcommand*\FB@@og{%
1324         \iflanguage{french}%
1325         {\ifFBAutoSpaceGuill\FB@og\ignorespaces
1326         \else\guillemotleft
1327         \fi}%
1328         {\guillemotleft}}%
1329     \AtBeginDocument{%
1330         \ifdefined\DeclareInputText
1331         \ifdefined\uc@dclc

```

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

```

1332         \uc@dclc{171}{default}{\FB@@og}%
1333     \else

```

if encoding is not utf8x, try utf8...

```

1334         \ifdefined\DeclareUnicodeCharacter
1335         utf8 loaded, use \DeclareUnicodeCharacter,

```

```

1335         \DeclareUnicodeCharacter{00AB}{\FB@@og}%
1336     \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.

```

1337         \@tempcnta'#1\relax
1338         \ifdefined\mule@def
1339         \mule@def{11}{\FB@@og}%
1340     \else
1341         \DeclareInputText{\the\@tempcnta}{\FB@@og}%
1342     \fi
1343 \fi
1344 \fi
1345 \else

```

Package inputenc not loaded, no way...

```

1346         \PackageWarning{frenchb.ldf}%
1347         {Option 'og' requires package inputenc.\MessageBreak}%
1348     \fi
1349 }%
1350 \fi
1351 }%

```

Same code for the closing quote.

```

1352 \define@key{FB}{fg}{%
1353     \ifFBunicode
1354         \ifFB@luatex@punct
1355             \FB@addGUILspace=1 \relax
1356         \fi
1357         \ifFB@xetex@punct
1358             \XeTeXcharclass"14 = \FB@guilf
1359             \XeTeXcharclass"BB = \FB@guilf
1360             \XeTeXcharclass"A0 = \FB@guilnul
1361             \XeTeXcharclass"202F = \FB@guilnul
1362         \fi
1363     \else
1364         \newcommand*{\FB@@fg}{%
1365             \iflanguage{french}%
1366                 {\ifFBAutoSpaceGuill\FB@fg
1367                 \else\guillemotright
1368                 \fi}%
1369             {\guillemotright}}%
1370     \AtBeginDocument{%
1371         \ifdefined\DeclareInputText
1372             \ifdefined\uc@dclc
1373                 \uc@dclc{187}{default}{\FB@@fg}%
1374             \else
1375                 \ifdefined\DeclareUnicodeCharacter
1376                     \DeclareUnicodeCharacter{00BB}{\FB@@fg}%
1377                 \else
1378                     \@tempcnta'#1\relax
1379                     \ifdefined\mule@def
1380                         \mule@def{27}{\FB@@fg}%
1381                     \else
1382                         \DeclareInputText{\the\@tempcnta}{\FB@@fg}%
1383                     \fi
1384                 \fi
1385             \fi
1386         \else
1387             \PackageWarning{frenchb.ldf}%
1388                 {Option 'fg' requires package inputenc.\MessageBreak}%
1389         \fi
1390     }%
1391 \fi
1392 }%
1393 }

```

\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.

```

1394 \newcommand*{\FBprocess@options}{%

```

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

```

1395 \@ifpackageloaded{enumitem}{%
1396   \ifFBStandardItemizeEnv
1397   \else
1398     \FBStandardItemizeEnvtrue
1399     \PackageInfo{frenchb.ldb}%
1400     {Setting StandardItemizeEnv=true for\MessageBreak
1401      compatibility with enumitem package,\MessageBreak}%
1402   \fi
1403   \ifFBStandardEnumerateEnv
1404   \else
1405     \FBStandardEnumerateEnvtrue
1406     \PackageInfo{frenchb.ldb}%
1407     {Setting StandardEnumerateEnv=true for\MessageBreak
1408      compatibility with enumitem package,\MessageBreak}%
1409   \fi}%
1410 \@ifpackageloaded{paralist}{%
1411   \ifFBStandardItemizeEnv
1412   \else
1413     \FBStandardItemizeEnvtrue
1414     \PackageInfo{frenchb.ldb}%
1415     {Setting StandardItemizeEnv=true for\MessageBreak
1416      compatibility with paralist package,\MessageBreak}%
1417   \fi
1418   \ifFBStandardEnumerateEnv
1419   \else
1420     \FBStandardEnumerateEnvtrue
1421     \PackageInfo{frenchb.ldb}%
1422     {Setting StandardEnumerateEnv=true for\MessageBreak
1423      compatibility with paralist package,\MessageBreak}%
1424   \fi}%
1425 \@ifpackageloaded{enumerate}{%
1426   \ifFBStandardEnumerateEnv
1427   \else
1428     \FBStandardEnumerateEnvtrue
1429     \PackageInfo{frenchb.ldb}%
1430     {Setting StandardEnumerateEnv=true for\MessageBreak
1431      compatibility with enumerate package,\MessageBreak}%
1432   \fi}%

```

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

```

1433 \def\FB@ufl{\update@frenchlists}
1434 \ifx\bbl@main@language\FB@french
1435   \update@frenchlists
1436 \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.

```
1437 \ifFBAutoSpacePunctuation
1438     \autospace@beforeFDP
1439 \else
1440     \noautospace@beforeFDP
1441 \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.

```
1442 \ifFBOriginalTypewriter
1443 \else
1444     \let\ttfamilyORI\ttfamily
1445     \let\rmfamilyORI\rmfamily
1446     \let\sffamilyORI\sffamily
1447     \let\ttfamily\ttfamilyFB
1448     \let\rmfamily\rmfamilyFB
1449     \let\sffamily\sffamilyFB
1450 \fi
```

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

```
1451 \ifFBThinColonSpace
1452     \ifFB@luatex@punct
1453         \FBcolonskip=\FBthinskip\relax
1454     \else
1455         \renewcommand*{\FBcolonspace}{\FBthinspace}%
1456     \fi
1457 \fi
```

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

```
1458 \ifFBINGuillSpace
1459     \ifFB@luatex@punct
1460         \FBguillskip=3.33pt plus 1.665pt minus 1.11pt \relax
1461     \else
1462         \renewcommand*{\FBguillspace}{\space}%
1463     \fi
1464 \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.

```
1465 \@ifpackageloaded{numprint}%
1466 {\ifnprt@autolanguage
1467     \providecommand*{\npstylefrench}{}%
1468     \ifFBThinSpaceInFrenchNumbers
1469         \renewcommand*\npstylefrench{%
1470             \npthousandsep{,}%

```

```

1471         \npdecimalsign{,}%
1472         \npproductsign{\cdot}%
1473         \npunitseparator{\,}%
1474         \npdegreeseperator{}%
1475         \nppercentseparator{\nprt@unitsep}%
1476     }%
1477 \else
1478     \renewcommand*\npstylefrench{%
1479         \npthousandsep{~}%
1480         \npdecimalsign{,}%
1481         \npproductsign{\cdot}%
1482         \npunitseparator{\,}%
1483         \npdegreeseperator{}%
1484         \nppercentseparator{\nprt@unitsep}%
1485     }%
1486 \fi
1487 \npaddtolanguage{french}{french}%
1488 \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.

```

1489 \ifFBFrenchSuperscripts
1490     \DeclareRobustCommand*\up*{\@ifstar{\FB@up@fake}{\fup}}%
1491 \else
1492     \DeclareRobustCommand*\up*{\@ifstar{\FB@up@fake}%
1493                                     {\textsuperscript}}%
1494 \fi

```

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

```

1495 \ifFBLowercaseSuperscripts
1496 \else
1497     \renewcommand*\FB@lc[1]{##1}%
1498 \fi

```

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

```

1499 \ifFBCustomiseFigTabCaptions
1500     \ifFB@koma
1501         \renewcommand*\captionformat{\CaptionSeparator}%
1502     \fi
1503     \@ifclassloaded{memoir}%
1504     {\captiondelim{\CaptionSeparator}}{}%
1505     \@ifclassloaded{beamer}%
1506     {\defbeamertemplate{caption label separator}{FBcustom}{%
1507         \CaptionSeparator}%
1508     \setbeamertemplate{caption label separator}{FBcustom}}{}%
1509 \else

```

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

```

1510 \ifFB@koma
1511 \renewcommand*{\captionformat}{\autospace@beforeFDP : }%
1512 \fi
1513 \@ifclassloaded{memoir}%
1514 {\captiondelim{\autospace@beforeFDP : }%
1515 }{}%
1516 \@ifclassloaded{beamer}%
1517 {\defbeamertemplate{caption label separator}{FBcolon}{%
1518 \autospace@beforeFDP : }%
1519 \setbeamertemplate{caption label separator}[FBcolon]%
1520 }{}%
1521 \fi

```

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

```

1522 \ifBShowOptions
1523 \GenericWarning{* }{%
1524 * **** List of possible options for frenchb ****\MessageBreak
1525 [Default values between brackets when frenchb is loaded *LAST*]%
1526 \MessageBreak
1527 ShowOptions=true [false]\MessageBreak
1528 StandardLayout=true [false]\MessageBreak
1529 GlobalLayoutFrench=false [true]\MessageBreak
1530 StandardLists=true [false]\MessageBreak
1531 IndentFirst=false [true]\MessageBreak
1532 ReduceListSpacing=false [true]\MessageBreak
1533 ListOldLayout=true [false]\MessageBreak
1534 StandardItemizeEnv=true [false]\MessageBreak
1535 StandardEnumerateEnv=true [false]\MessageBreak
1536 StandardItemLabels=true [false]\MessageBreak
1537 ItemLabels=\textemdash, \textbullet,
1538 \protect\ding{43},... [\textendash]\MessageBreak
1539 ItemLabeli=\textemdash, \textbullet,
1540 \protect\ding{43},... [\textendash]\MessageBreak
1541 ItemLabelii=\textemdash, \textbullet,
1542 \protect\ding{43},... [\textendash]\MessageBreak
1543 ItemLabeliii=\textemdash, \textbullet,
1544 \protect\ding{43},... [\textendash]\MessageBreak
1545 ItemLabeliv=\textemdash, \textbullet,
1546 \protect\ding{43},... [\textendash]\MessageBreak
1547 FrenchFootnotes=false [true]\MessageBreak
1548 AutoSpaceFootnotes=false [true]\MessageBreak
1549 AutoSpacePunctuation=false [true]\MessageBreak
1550 OriginalTypewriter=true [false]\MessageBreak
1551 ThinColonSpace=true [false]\MessageBreak
1552 ThinSpaceInFrenchNumbers=true [false]\MessageBreak
1553 FrenchSuperscripts=false [true]\MessageBreak
1554 LowercaseSuperscripts=false [true]\MessageBreak
1555 PartNameFull=false [true]\MessageBreak
1556 SuppressWarning=true [false]\MessageBreak
1557 CustomiseFigTabCaptions=false [true]\MessageBreak
1558 OldFigTabCaptions=true [false]\MessageBreak

```

```

1559 SmallCapsFigTabCaptions=false [true]\MessageBreak
1560 INGuillSpace=true [false]\MessageBreak
1561 InnerGuillSingle=true [false]\MessageBreak
1562 EveryParGuill=open, close, none [open]\MessageBreak
1563 EveryLineGuill=open, close, none
1564         [open in LuaTeX, none otherwise]\MessageBreak
1565 og= <left quote character>, fg= <right quote character>%
1566 \MessageBreak
1567 *****%
1568 \MessageBreak\protect\frenchbsetup{ShowOptions}}
1569 \fi
1570 }

```

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.

```

1571 \AtBeginDocument{%
1572   \providecommand*\xspace{\relax}%
1573   \ifdefined\pdfstringdefDisableCommands
1574     \pdfstringdefDisableCommands{%
1575       \let\up\relax
1576       \let\fu\relax
1577       \let\degre\textdegree
1578       \let\degres\textdegree
1579       \def\ieme{e\xspace}%
1580       \def\iemes{es\xspace}%
1581       \def\ier{er\xspace}%
1582       \def\iers{ers\xspace}%
1583       \def\iere{re\xspace}%
1584       \def\ieres{res\xspace}%
1585       \def\FrenchEnumerate#1{#1\degre\space}%
1586       \def\FrenchPopularEnumerate#1{#1\degre)\space}%
1587       \def\No{N\degre\space}%
1588       \def\no{n\degre\space}%
1589       \def\Nos{N\degre\space}%
1590       \def\nos{n\degre\space}%
1591       \def\FB@og{\guillemotleft\space}%
1592       \def\FB@fg{\space\guillemotright}%
1593       \def\at{@}%
1594       \def\circonflexe{\string^}%
1595       \def\tild{\string~}%
1596       \let\bsc\textsc
1597     }%
1598   \fi

```

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

```

1599 \FBprocess@options

```


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

```
1600 \ifFB@luatex@punct
1601 \activate@luatexpunct
1602 \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!

```
1603 \ifFBunicode
1604 \ifdefined\DeclareUTFcharacter
1605 \else
1606 \ifpackageloaded{luainputenc}{}%
1607 {\PackageWarning{frenchb.ldf}%
1608 {Add \protect\usepackage{fontspec} to the\MessageBreak
1609 preamble of your document,}%
1610 }%
1611 \fi
1612 \else
1613 \begingroup \newcommand{\FBOTone}{OT1}%
1614 \ifx\encodingdefault\FBOTone
1615 \PackageWarning{frenchb.ldf}%
1616 {OT1 encoding should not be used for French.%
1617 \MessageBreak
1618 Add \protect\usepackage[T1]{fontenc} to the
1619 preamble\MessageBreak of your document,}%
1620 \fi
1621 \endgroup
1622 \fi
1623 }
```

2.11 French lists

`\listFB` Vertical spacing in lists should be shorter in French texts than the defaults provided `\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.

```
1624 \let\listORI\list
```

```

1625 \let\endlistORI\endlist
1626 \def\FB@listVsettings{%
1627     \setlength{\itemsep}{0.4ex plus 0.2ex minus 0.2ex}%
1628     \setlength{\parsep}{0.4ex plus 0.2ex minus 0.2ex}%
1629     \setlength{\topsep}{0.8ex plus 0.4ex minus 0.4ex}%
1630     \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.
1631     \@tempdima=\parskip
1632     \addtolength{\topsep}{-\@tempdima}%
1633     \addtolength{\partopsep}{\@tempdima}%
1634 }
1635 \def\listFB#1#2{\listORI{#1}{\FB@listVsettings #2}}
1636 \let\endlistFB\endlist

```

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

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

Text starting at 'parindent' \Leftarrow Leftmargin — first item... — first second level item — next one... — second item...

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

```

\FrenchLabelItem \Frlabelitemi \newcommand*{\FrenchLabelItem}{\textemdash}
\FrenchLabelItem \Frlabelitemii \newcommand*{\Frlabelitemii}{\FrenchLabelItem}
\FrenchLabelItem \Frlabelitemiii \newcommand*{\Frlabelitemiii}{\FrenchLabelItem}
\FrenchLabelItem \Frlabelitemiv \newcommand*{\Frlabelitemiv}{\FrenchLabelItem}
\FrenchLabelItem \Frlabelitemv \newcommand*{\Frlabelitemv}{\FrenchLabelItem}

```

`\listindentFB` Let's define two lengths `\listindentFB` and `\labelwidthFB` to customise lists' horizontal indentations. They are given silly values here (–1pt) 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.

```

1642 \newlength\listindentFB
1643 \setlength{\listindentFB}{-1pt}
1644 \newlength\labelwidthFB
1645 \setlength{\labelwidthFB}{-1pt}

```

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

```

1646 \newlength\leftmarginFB
1647 \def\FB@listHsettings{%
1648   \leftmarginFB\labelwidthFB
1649   \advance\leftmarginFB \labelsep
1650   \leftmargini\leftmarginFB
1651   \advance\leftmargini \listindentFB
1652   \leftmarginii\leftmarginFB
1653   \leftmarginiii\leftmarginFB
1654   \leftmarginiv\leftmarginFB
1655   \leftmargin\csname leftmargin\romannumeral\the\@listdepth\endcsname
1656 }

```

`\itemizeFB` New environment for French itemize-lists.

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

```

1657 \def\FB@itemizesettings{%
1658   \ifFBReduceListSpacing
1659     \setlength{\itemsep}{\z@}%
1660     \setlength{\parsep}{\z@}%
1661     \setlength{\topsep}{\z@}%
1662     \setlength{\partopsep}{\z@}%
1663     \@tempdima=\parskip
1664     \addtolength{\topsep}{-\@tempdima}%
1665     \addtolength{\partopsep}{-\@tempdima}%
1666   \fi
1667   \settowidth{\labelwidth}{\csname\@itemitem\endcsname}%
1668   \ifFBListOldLayout
1669     \setlength{\leftmargin}{\labelwidth}%
1670     \addtolength{\leftmargin}{\labelsep}%
1671     \addtolength{\leftmargin}{\parindent}%
1672   \else
1673     \FB@listHsettings
1674   \fi
1675 }

```

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`.

```

1676 \def\itemizeFB{%
1677   \ifnum \@itemdepth >\thr@@\toodeep\else
1678     \advance\@itemdepth\@ne
1679     \edef\@itemitem{labelitem\romannumeral\the\@itemdepth}%
1680     \expandafter
1681     \listORI
1682     \csname\@itemitem\endcsname
1683     \FB@itemizesettings

```

```

1684 \fi
1685 }
1686 \let\enditemizeFB\endlistORI

1687 \def\labelitemsFB{%
1688 \let\labelitemi\Frlabelitemi
1689 \let\labelitemii\Frlabelitemii
1690 \let\labelitemiii\Frlabelitemiii
1691 \let\labelitemiv\Frlabelitemiv
1692 \ifdim\labelwidthFB<\z@
1693 \settowidth{\labelwidthFB}{\FrenchLabelItem}%
1694 \fi
1695 \ifdim\listindentFB<\z@
1696 \ifdim\parindent=\z@
1697 \setlength{\listindentFB}{1.5em}%
1698 \else
1699 \setlength{\listindentFB}{\parindent}%
1700 \fi
1701 \fi
1702 }

```

\enumerateFB The definition of `\enumerateFB`, new to version 2.6a, follows the one of `\enumerate` in standard $\text{\LaTeX} 2_{\varepsilon}$ 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`.

```

1703 \def\enumerateFB{%
1704 \ifnum \@enumdepth >\thr@@\toodeep\else
1705 \advance\@enumdepth\@ne
1706 \edef\@enumctr{enum\romannumeral\the\@enumdepth}%
1707 \expandafter
1708 \list
1709 \csname label\@enumctr\endcsname
1710 {\FB@listHsettings
1711 \usecounter\@enumctr\def\makelabel##1{\hss\llap{##1}}}%
1712 \fi
1713 }
1714 \let\endenumerateFB\endlistORI

```

\descriptionFB Same tuning for the description environment (see the original definition in `classes.dtx`). Customisable `\listindentFB` added to `\itemindent` (first level only).

```

1715 \def\descriptionFB{%
1716 \list{}{\FB@listHsettings
1717 \labelwidth\z@
1718 \itemindent-\leftmargin
1719 \ifnum\@listdepth=1
1720 \advance\itemindent by \listindentFB
1721 \fi
1722 \let\makelabel\descriptionlabel}%
1723 }
1724 \let\enddescriptionFB\endlistORI

```

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

`\bbl@frenchlistlayout` 1725 `\def\update@frenchlists{%`
`\bbl@nonfrenchlistlayout` 1726 `\ifFBReduceListSpacing \let\list\listFB \fi`
1727 `\ifFBStandardItemizeEnv`
1728 `\else \let\itemize\itemizeFB \fi`
1729 `\ifFBStandardItemLabels`
1730 `\else \labelitemsFB \fi`
1731 `\ifFBStandardEnumerateEnv`
1732 `\else \let\enumerate\enumerateFB \let\description\descriptionFB \fi`
1733 `}`

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. 52.

1734 `\def\FB@ufl{\relax}`
1735 `\def\bbl@frenchlistlayout{%`
1736 `\ifFBGlobalLayoutFrench`
1737 `\else`
1738 `\babel@save\list \babel@save\itemize`
1739 `\babel@save\enumerate \babel@save\description`
1740 `\babel@save\labelitemi \babel@save\labelitemii`
1741 `\babel@save\labelitemiii \babel@save\labelitemiv`
1742 `\fi`
1743 `\FB@ufl`
1744 `}`
1745 `\def\bbl@nonfrenchlistlayout{%`
1746 `\ifFBGlobalLayoutFrench`
1747 `\update@frenchlists`
1748 `\fi`
1749 `}`
1750 `\FB@addto{extras}{\bbl@frenchlistlayout}`
1751 `\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`.
`\bbl@nonfrenchindent` We will need to save the value of the flag `\if@afterindent` ‘AtBeginDocument’ before eventually changing its value.

1752 `\def\bbl@frenchindent{%`
1753 `\ifFBGlobalLayoutFrench\else\babel@save\@afterindentfalse\fi`
1754 `\ifFBIndentFirst`
1755 `\let\@afterindentfalse\@afterindenttrue`
1756 `\@afterindenttrue`
1757 `\fi}`
1758 `\def\bbl@nonfrenchindent{%`
1759 `\ifFBGlobalLayoutFrench`
1760 `\ifFBIndentFirst`
1761 `\@afterindenttrue`

```

1762     \fi
1763   \fi}
1764 \FB@addto{extras}{\bbl@frenchindent}
1765 \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 `\ifFBAutoSpaceFootnotes` 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 `\ifFBAutoSpaceFootnotes`.

```

1766 \AtBeginDocument{\@ifpackageloaded{bigfoot}%
1767                   {\PackageInfo{frenchb.ldb}%
1768                     {bigfoot package in use.\MessageBreak
1769                     frenchb will NOT customise footnotes;\MessageBreak
1770                     reported}}%
1771                   {\let\@footnotemarkORI\@footnotemark
1772                     \def\@footnotemarkFB{\leavevmode\unskip\unkern
1773                                           \,\@footnotemarkORI}%
1774                     \ifFBAutoSpaceFootnotes
1775                       \let\@footnotemark\@footnotemarkFB
1776                     \fi}%
1777                   }

```

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).

```

1778 \newdimen\parindentFFN
1779 \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.

```

1780 \newcommand*{\dotFFN}{.}
1781 \newcommand*{\kernFFN}{\kern .5em}
1782 \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.

```

1783 \ifFB@koma
1784   \let\@makefntextORI\@makefntext
1785   \let\@@makefnmarkORI\@@makefnmark

```

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

```

1786   \deffootnote[\FBfnindent]{0pt}{\parindentFFN}%
1787           {\thefootnotemark\dotFFN\kernFFN}
1788   \let\@makefntextFB\@makefntext
1789   \let\@@makefnmarkFB\@@makefnmark

```

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

```

1790   \deffootnote[\parindentFFN]{0pt}{\parindentFFN}%
1791           {\textsuperscript{\thefootnotemark}}
1792   \let\@makefntextTH\@makefntext
1793   \let\@@makefnmarkTH\@@makefnmark

```

Restore the original definitions.

```

1794   \let\@makefntext\@makefntextORI
1795   \let\@@makefnmark\@@makefnmarkORI
1796 \fi

```

Definitions for the memoir class:

```

1797 \ifclassloaded{memoir}

```

(see original definition in memman.pdf)

```

1798   {\newcommand{\@makefntextFB}[1]{%
1799     \def\footscript##1{##1\dotFFN\kernFFN}%
1800     \setlength{\footmarkwidth}{\FBfnindent}%
1801     \setlength{\footmarksep}{-\footmarkwidth}%
1802     \setlength{\footparindent}{\parindentFFN}%
1803     \makefootmark #1}%
1804   }}

```

Definitions for the beamer class:

```

1805 \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.

```

1806   {\def\@makefntextFB#1{%
1807     \def\insertfootnotetext{#1}%
1808     \def\insertfootnotemark{\insertfootnotemarkFB}%
1809     \usebeamertemplate***{footnote}}%

```

```

1810 \def\insertfootnotemarkFB{%
1811 \usebeamercolor[fg]{footnote mark}%
1812 \usebeamerfont*{footnote mark}%
1813 \@thefnmark\dotFFN\kernFFN}%
1814 }{}

```

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

```

1815 \providecommand*\insertfootnotemarkFB{%
1816 \parindent=\parindentFFN
1817 \rule\z@\footnotesep
1818 \setbox\@tempboxa\hbox{\@thefnmark}%
1819 \ifdim\wd\@tempboxa>\z@
1820 \llap{\@thefnmark}\dotFFN\kernFFN
1821 \fi}
1822 \providecommand\@makefntextFB[1]{\insertfootnotemarkFB #1}

```

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

```

1823 \AtBeginDocument{%
1824 \@ifpackageloaded{bigfoot}{}%
1825 {\ifdim\parindentFFN<10in
1826 \else
1827 \parindentFFN=\parindent
1828 \ifdim\parindentFFN<1.5em \parindentFFN=1.5em \fi
1829 \fi
1830 \settowidth{\FBfnindent}{\dotFFN\kernFFN}%
1831 \addtolength{\FBfnindent}{\parindentFFN}%
1832 \let\@makefntextORI\@makefntext
1833 \ifFB@koma

```

Definition of \@makefntext for koma-script classes:

```

1834 \let\@@makefnmarkORI\@@makefnmark
1835 \long\def\@makefntext#1{%
1836 \ifFBFrenchFootnotes
1837 \ifx\footnote\thanks
1838 \let\@@makefnmark\@@makefnmarkTH
1839 \@makefntextTH{#1}%
1840 \else
1841 \let\@@makefnmark\@@makefnmarkFB
1842 \@makefntextFB{#1}%
1843 \fi
1844 \else
1845 \let\@@makefnmark\@@makefnmarkORI
1846 \@makefntextORI{#1}%
1847 \fi}%
1848 \else

```


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

```

1849      \@ifclassloaded{memoir}%
1850      {\ifFBFrenchFootnotes
1851        \setlength{\thanksmarkwidth}{\parindentFFN}%
1852        \setlength{\thanksmarksep}{-\thanksmarkwidth}%
1853        \fi
1854      }{}%
```

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

```

1855      \@ifclassloaded{beamer}%
1856      {\ifFBFrenchFootnotes
1857        \ifdim\parindentFFN=1.5em\else
1858          \FBWarning{frenchb}{%
1859            \protect\parindentFFN\space is ineffective%
1860            \MessageBreak within the beamer class.\MessageBreak
1861            Reported}%
1862          \fi
1863        \fi
1864      }{}%
```

Definition of `\@makefn`text for all classes other than koma-script:

```

1865      \long\def\@makefn#1{%
1866        \ifFBFrenchFootnotes
1867          \@makefnFB{#1}%
1868        \else
1869          \@makefnORI{#1}%
1870        \fi}%
1871      \fi
1872    }%
1873 }
```

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 `\@makefn`text.

```

1874 \newcommand*{\AddThinSpaceBeforeFootnotes}{\FBAutoSpaceFootnotesttrue}
1875 \newcommand*{\FrenchFootnotes}{\FBFrenchFootnotesttrue}
1876 \newcommand*{\StandardFootnotes}{\FBFrenchFootnotestfalse}
```

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.

```

1877 \FBcleanon@exit
```

1878 \let\FB@llc\loadlocalcfg
1879 \let\loadlocalcfg\@gobble
1880 \ldf@finish\CurrentOption
1881 \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).	43	
\bsc: \hbox dropped, replaced by \kern0pt.	36	
\captionsfrench: 'Fig.' changed to 'Figure' and 'Tab.' to 'Table'. . . .	39	
\datefrench: 2 '\relax' added in \today's definition.	32	
\nombre: \nombre now requires numprint.sty.	38	
General: \parindentFFN not changed if already defined (required by JA for cah-gut.cls).	62	
Added warning for OT1 encoding.	56	
Footnotes are now printed by default 'à la française' for the whole document.	62	
New command \frenchbsetup added for global customisation. . .	45	
v2.0b		
General: Footnotes: Just do nothing (except warning) when the bigfoot package is loaded.	62	
v2.0c		
\frenchbsetup: Option ThinSpaceInFrenchNumbers added.	45	
General: There is no need to define here numprint's command \npstylefrench, it will be redefined 'AtBeginDocument' by \FBprocess@options.	39	
v2.0d		
\frenchbsetup: Options og and fg changed: limit the definition to French so that quote characters can be used in German.	45	
v2.0e		
\frenchbsetup: New option: StandardLists.	45	
v2.0f		
\frenchbsetup: StandardLayout option had no effect on lists. Test moved to \FBprocess@options. . . .	45	
Two typos corrected in option StandardLists: [false] → [true]		
		and StandardLayout → StandardLists. 45
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. . .	45	
v2.1a		
\datefrench: \today changed (correction in 2.0 was wrong: \today was printed without spaces in toc).	32	
\frenchbsetup: New option: FrenchSuperscripts to define \up as \fup or as \textsuperscript. . . .	45	
New option: LowercaseSuperscripts.	45	
General: Command \fup added to produce better superscripts than \textsuperscript.	33	
v2.1b		
\fup: Command \fup changed to use real superscripts from fourier v. 1.6.	33	
General: Disable some commands in bookmarks.	56	
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).	37	
\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).	33	
General: Added commands \Nos and \nos.	35	

v2.1d	General: In LaTeX, frenchb no longer adds spaces before ‘high punctuation’ characters in computer code. Suggested by Yannis Haralambous.	27
	General: Argument of <code>\ProvidesLanguage</code> changed above from ‘french’ to ‘frenchb’ (otherwise <code>\listfiles</code> prints no date/version information). The real name of current language (french) as to be corrected before calling <code>\LdfInit</code>	11
	Avoid warning “\end occurred when <code>\ifx ... incomplete</code> ” with LaTeX-2.09.	11
v2.2a	<code>\frenchbsetup</code> : Default values of flags changed: default now means ‘StandardLayout’, they will be changed to ‘FrenchLayout’ <code>AtEndOfPackage</code> only if french is <code>\bbl@main@language</code>	45
	The global layout of the document is no longer changed when frenchb is not the last option of babel (<code>\bbl@main@language</code>). Suggested by Ulrike Fischer.	45
	When frenchb is babel’s last option, French becomes the document’s main language, so <code>GlobalLayoutFrench</code> applies.	45
	<code>\fup</code> : <code>\newif</code> and <code>\newdimen</code> moved before <code>\ifLaTeXe</code> to avoid an error with plainTeX.	33
v2.3a	<code>\NoAutoSpaceBeforeFDP</code> : <code>\NoAutoSpaceBeforeFDP</code> and <code>\AutoSpaceBeforeFDP</code> now set the flag <code>\ifFBAutoSpacePunctuation</code> accordingly (LaTeX only).	27
	<code>\frenchbsetup</code> : New option: <code>OriginalTypewriter</code> . Now frenchb switches to <code>\noautospace@beforeFDP</code> when a tt-font is in use. When <code>OriginalTypewriter</code> is set to true, frenchb behaves as in pre-2.3 versions.	45
	<code>\fup</code> : <code>\lowercase</code> changed to <code>\MakeLowercase</code> as the former doesn’t work for non ASCII characters in encodings like <code>applemac</code> , <code>utf-8</code> ,...	33
v2.3b	General: New commands <code>\dotFFN</code> and <code>\kernFFN</code> for more flexibility (suggested by JA).	62
v2.3c	<code>\ttfamilyFB</code> : Commands <code>\ttfamily</code> , <code>\rmfamily</code> and <code>\sffamily</code> have to be robust. Bug introduced in 2.3a, pointed out by Manuel Pégourié-Gonnard.	27
v2.3d	<code>\bbl@nonfrenchindent</code> : Bug correction: previous versions of frenchb set the flag <code>\if@afterindent</code> to false outside French which is correct for English but wrong for some languages like Spanish. Pointed out by Juan José Torrens.	61
v2.3e	<code>\NoAutoSpaceBeforeFDP</code> : Execute <code>\AutoSpaceBeforeFDP</code> also in LaTeX to define <code>\FDP@colonspace</code> : needed for <code>tex4ht</code> , pointed out by MPG.	27
v2.4a	<code>\CaptionSeparator</code> : <code>\PackageWarning</code> changed to <code>\FBWarning</code> (in case <code>\@makecaption</code> has been customised). <code>\FBWarning</code> is defined as <code>\PackageWarning</code> by default but can be made silent using <code>\frenchbsetup</code> , (suggested by MPG).	41
	<code>\frenchbsetup</code> : New option <code>SuppressWarning</code>	45
	<code>\ifFBXeTeX</code> : Added a new ‘if’ <code>\FBunicode</code> and some <code>\lccode</code> definitions to <code>\extrasfrench</code> and <code>\noextrasfrench</code>	13
	General: <code>\PackageWarning</code> changed to <code>\FBWarning</code> (when bigfoot package in use).	62
v2.4c	<code>\ttfamilyFB</code> : In <code>\ttfamilyFB</code> , also	

cancel automatic spaces inside French guillemets entered as characters (see <code>\frenchbsetup</code>).	27	available for French. In this case <code>\l@french</code> and <code>\l@english</code> are 0. Pointed out by Günter Milde.	46
General: In <code>\ttfamilyFB</code> , also cancel automatic spaces inside French guillemets coded as characters (see <code>\frenchbsetup</code>).	49	v2.5c General: The code meant for XeTeX also works for LuaTeX, we just need to change the test.	49
v2.4d <code>\up</code> : Command <code>\up</code> defined with <code>\providecommand</code> instead of <code>\newcommand</code> as <code>\up</code> may be defined elsewhere (<code>catalan.ldf</code>). Bug pointed out by Felip Manyé i Ballester.	33	v2.5d <code>\ifFBXeTeX</code> : Added two new ‘if’ <code>\FBXeTeX</code> and <code>\FBLuaTeX</code> as XeTeX and behave differently regarding the status of the French “apostrophe”.	13
v2.5a <code>\FBthinspace</code> : Define <code>\FBthinspace</code> for those who want to customise the width of the space before ; and co.	15	General: Moved the <code>\newcount</code> command outside <code>\ifFB@xetex@punct ... \fi</code> (it broke Plain formats).	23
<code>\captionsfrench</code> : <code>\emph</code> deleted in <code>\seename</code> and <code>\alsoname</code> to match what is done for the other languages. Suggested by Marc Baudoin.	39	v2.5e General: <code>\pdfstringdefDisableCommands</code> should redefine <code>\FB@og</code> and <code>\FB@fg</code> instead of <code>\og</code> and <code>\fg</code> so that it works also when quotes are entered as characters. Reported by Sébastien Gouezel.	56
<code>\fg</code> : <code>\og</code> and <code>\fg</code> do not print correctly in English when using XeTeX or LuaTeX, fixed by using <code>\textquotedblleft</code> and <code>\textquotedblright</code> defined above.	30	v2.5f <code>\FBtextellipsis</code> : Unicode fonts also provide a ready made character for <code>\textellipsis</code> , let’s just use it (reported by Maxime Chupin, 2011/06/04).	43
<code>\textquoteddblright</code> : Change <code>\guillemotleft</code> and <code>\guillemotright</code> definitions for Unicode and provide definitions for <code>\textquotedblleft</code> and <code>\textquotedblright</code> . Insures correct printing of quotes by <code>\og</code> and <code>\fg</code> in French and outside.	28	General: Changed definitions of <code>\at</code> , <code>\circonflexe</code> , <code>\tild</code> , <code>\boi</code> and <code>\degre</code> for Unicode based engines.	36
General: New command <code>\NoAutoSpacing</code> , suggested by MPG.	28	v2.5g <code>\FB@xetex@punct@french</code> : XeTeXcharclass(es) for French quotes will be set to <code>\FB@guilo</code> and <code>\FB@guilf</code> by options ‘og’ and ‘fg’ in <code>\frenchbsetup</code> . French quotes should behave as normal characters by default in XeLaTeX as in LaTeX.	23
Punctuation is no longer made active with XeTeX-based engines.	14	General: Redefine <code>\degre</code> , <code>\degres</code> <code>\at</code> <code>\circonflexe</code> and <code>\tild</code> for bookmarks. Add <code>\fup</code> also.	56
v2.5b General: Do not use the test <code>\iflanguage{french}</code> to check whether French is the main language or not, as it might be erroneously positive when English is the main language and no hyphenation patterns are		When <code>\ifFB@xetex@punct</code> is true, ‘og’ and ‘fg’ options now set XeTeXcharclasses of these characters to <code>\FB@guilo</code> and	

own space before ‘high punctuation’ characters.	23	General: \LdfInit checks \datefrench instead of \captionsfrench to avoid a conflict with papertex.cls which loads datetime.sty.	11
v2.6h		\bbl@nonfrenchguillemets deleted, use \babel@save instead.	30
\CaptionSeparator: No active catcodes in \STD@makecaption’s definition.	41	french.cfg will be loaded (if found) instead of frenchb.cfg. NO NEED for .cfg files in French anyway. ..	65
General: \FG@og and \FG@fg changed: former clumsy code removed. ..	49	In Plain, provide a substitute for \PackageWarning and \PackageInfo.	12
If \@makecaption is undefined, no warning.	42	Merging of \captionsfrenchb, \captionsfrançais with \captionsfrench deleted in favor of new babel 3.9 syntax.	40
New class \FB@guilnul for characters U+00A0 (Unicode nobreakspace) and U+202F (Unicode nobreakthinspace), to prevent frenchb from adding spurious spaces inside quotes. ..	23	More informative, less TeXnical warning about \@makecaption. .	43
v3.0a		New flag \ifFB@luatex@punct for ‘high punctuation’ management with LuaTeX engines.	14
\CaptionSeparator: Remove \CaptionSeparatorORI, use \babel@save instead.	41	New handling of ‘high punctuation’ through callbacks with LuaTeX engines.	16
\FB@fg: Added explicit \FBguillskip for LuaTeX.	29	No warning about \@makecaption for SMF classes. No warning either with LuaTeX or XeTeX engines. ..	42
Definitions of \FB@og and \FB@fg now depend on punctuation handling (LuaTeX / XeTeX / active). ..	29	Options processing completely reorganised, now \babel@save and \babel@savevariable are usable for French.	45
\FBprocess@options: Changed option ThinColonSpace to make it work also with LuaTeX.	53	Support for options frenchb, français, canadien, acadian changed.	11
With koma-script and memoir class, customise \captionformat and \captiondelim.	54	Test \ifXeTeX changed to \ifFBunicode and ‘xltxtra’ changed to ‘fontspec’.	57
\FBthinskip: LuaTeX requires dimensions: two new skips \FBcolonskip and \FBthinskip. ..	15	v3.0b	
\captionsfrench: Take advantage of babel’s \SetString commands for captionnames.	39	General: frenchb.lua was not found by Lua function dofile (not kpathsea aware). Call function kpse.find_file first, as suggested by Paul Gaborit.	22
\datefrench: Take advantage of babel’s \SetString commands for \datefrench. Doesn’t work with Plain (yet?).	32	Require luatexbase with LaTeX in case fontspec has not been loaded before babel.	16
\descriptionFB: Add \listindentFB to \itemindent. Suggested by Denis Bitouzé. ...	60	v3.0c	
\extrasfrench: Take advantage of babel’s \babel@savevariable to handle apostrophe’s \lccode. ..	13	\FB@fg: Changed \FBguill@spacing (internal) to \FBguillspace	
\frenchbsetup: New options OldFigTabCaptions and CustomiseFigTabCaptions.	45		

(public).	29	\fprimo): Removed \lowercase from definitions of \FrenchEnumerate, ... \No and co: \up already does the conversion.	35
\datefrench: \SetString still does not work for Plain with babel 3.9k. Need to define \datefrench. . . .	32	\frenchbsetup: New option	
\frenchbsetup: New option		INGuillSpace.	45
General: Activate option StandardLists when beamer class is loaded. . .	46	\frenchbsetup: New option	
frenchb requires babel-3.9i.	12	SmallCapsFigTabCaptions.	45
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.	19	\ieres: Removed \lowercase from definitions of \ieme and co: \up already does the conversion. . . .	35
Just load luatexbase.sty instead of luaotfload.sty with plain formats. .	16	General: frenchb.lua: add a check for null fid in french_punctuation (Tikz \nullfont). Bug pointed out by Paul Gaborit.	18
No need to define \l@french as \lang@french, babel.def (3.9j) takes care for this.	11	v3.1c	
v3.1a		General: frenchb.lua: Previous bug fix for null glues (v3.0c) did not work properly. Fixed now (I hope). Pointed out by Jacques André. . .	19
\frenchbsetup: New options		v3.1d	
InnerGuillSingle, EveryParGuill and EveryLineGuill to control \frquote.	45	General: New section: issue warnings if packages listings, numprint and natbib are loaded too early or too late vs babel.	44
General: Codes "13 and "14 added for French quotes in T1-encoding. Support for older versions of LuaTeX and XeTeX dropped. . . .	49	v3.1e	
fontspec is not required for T1 fonts used with the luainputenc.sty package.	57	\frenchbsetup: Corrected typo: SmallCapsFigTabCaptions instead of SmallCapsFigTabCaptions. Pointed out by Céline Chevalier. .	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}»).	20	v3.1f	
frenchb.lua: codes 0x13 and 0x14 added for French quotes in T1-encoding.	16	\FBprocess@options: Bug fix for the beamer class: figure and table captions are now consistent with frenchb's documentation. Pointed out by Denis Bitouzé.	54
frenchb.lua: look ahead when next is a kern (i.e. in «\texttt{a} »). .	20	Definition of \captionformat and \captiondelim changed when option CustomiseFigTabCaptions is set to false.	54
Misplaced \fi for plain formats. .	16	\FBthinspace: \FBthinspace is no longer a kern but a skip (frenchb adds a nobreak penalty before it). .	15
New command \frquote for imbedded or long French quotations.	30	General: \FBCaption@Separator changed when option CustomiseFigTabCaptions is set to false.	43
v3.1b		v3.1g	
\captionsfrench: Change \scshape to customisable \FBfigtabshape for \figurename and \tablename.	39	\captionsfrench: \partname's definition depends now on flag PartNameFull. No need to redefine it in \frenchbsetup.	39

Bug fix for koma-scripts classes: a spurious dot was added by the <code>\partformat</code> command.	40	group in order to ensure compatibility with package <code>wrapfig</code>	31
<code>\frenchbsetup</code> : <code>PartNameFull</code> now just sets the flag, nothing to add to <code>\captionfrench</code> when false.	45	General: Loading <code>luatexbase.sty</code> is no longer needed with LaTeX release 2015/10/01 or later.	16
General: <code>frenchb.lua</code> : flag <code>addgl</code> set to false for '«' at the end of an <code>\hbox</code> or a paragraph or when followed by a null glue (i.e. springs).	20	v3.1k <code>\FB@xetex@punct@french</code> : Thin glues (less than 1sp) should not trigger space insertion before high punctuation. Add a check on <code>\lastkip</code>	23
<code>frenchb.lua</code> : flag <code>addgl</code> set to false for '»' at the beginning of an <code>\hbox</code> or a paragraph or a tabular 'l' and 'c' columns.	20	General: (pdfTeX shorthands) test on <code>\lastskip</code> changed from 0pt to 1sp for active punctuation for consistency with XeTeX and LuaTeX.	25
<code>frenchb.lua</code> : node <code>HLIST</code> added; node <code>TEMP</code> added for the first node of <code>\hboxes</code>	17	v3.1l <code>\FB@luatex@punct@french</code> : Use <code>\babel@save</code> to save and restore <code>\shorthandon</code> and <code>\shorthandoff</code>	21
Lua function <code>french_punctuation</code> is now inserted at the end of the "kerning" callback (no priority) instead of <code>"hpack_filter"</code> and <code>"pre_linebreak_filter"</code>	22	<code>\FB@xetex@punct@french</code> : Save and restore <code>\XeTeXinterchartokenstate</code> , <code>\shorthandon</code> , <code>\shorthandoff</code> using <code>\babel@savevariable</code> and <code>\babel@save</code> , <code>\XeTeXcharclass(es)</code> using <code>\FB@savevariable@loop</code>	23
Use Babel defined loops <code>\bbl@for</code> instead of <code>\@for</code> borrowed from file <code>ltnctrl.dtx</code> (<code>\@for</code> is undefined in Plain).	23	General: Add a variant of <code>\babel@savevariable</code> to save <code>\XeTeXcharclass(es)</code> in a loop.	23
v3.1h General: <code>french.cfg</code> from e-french conflicts with <code>frenchb</code> . Do NOT load it (no need for .cfg files with <code>frenchb</code> anyway).	65	<code>frenchb.lua</code> : <code>font.getfont(fid)</code> possibly returns nil even for a positive fid (i.e. <code>AMS lcircle1.pfb</code>). Reported by François Legendre.	18
v3.1i <code>\frquote</code> : <code>\luatexlocalleftbox</code> changed to <code>\localleftbox</code> by new LaTeX release 2015/10/01.	31	v3.1m General: <code>frenchb.lua</code> : <code>new_glue_scaled</code> returns nil in case of invalid font table (i.e. <code>lcircle1.pfb</code>). In such cases <code>frenchb</code> leaves the node list unchanged.	18
General: <code>\nombre</code> command changed when <code>numprint.sty</code> is not loaded: only one warning, no error.	38	v3.2a <code>\fg</code> : <code>\xspace</code> moved from <code>\FB@fg</code> to <code>\fg</code> : <code>\xspace</code> messes up <code>\frquote</code> , pointed out by Sonia Labetoulle. As a side effect <code>\xspace</code> is now active in <code>\fg</code> in and outside French.	30
Compatibility code added due to changes in the 2015/10/01 LaTeX release, see <code>ltnews23.tex</code>	16		
Remove restriction about loading <code>numprint.sty</code> after <code>babel</code>	44		
v3.1j <code>\frquote</code> : <code>\PackageWarning</code> is undefined in Plain, use <code>\fb@warning</code> instead.	31		
<code>\frquote</code> completely rewritten: <code>\leavevmode</code> added and explicitly save/retore <code>\everypar</code> and <code>\localleftbox</code> instead of using a			

General: beamer.cls requires a specific definition of <code>\@makefnctextFB</code> (pointed out by DB). The same is true for memoir and koma-script classes (done). .	62	characters: it's value will be 4095 for new engines and 255 for older ones.	15
v3.2b		General: frenchb.lua: <code>glue_spec</code> removed; starting with LuaTeX 0.95, glue specifications fit in glue.	18
<code>\NoAutoSpacing</code> :		Load <code>lualatex.tex</code> for plain LuaTeX to ensure <code>\newattribute</code> is defined.	16
<code>\noautospace@beforeFDP</code> added to <code>\NoAutoSpacing</code> in case of active punctuation.		Warning added when the subcaption package is loaded before babel/frenchb.	42
<code>\NoAutoSpacing</code> made robust. .	28		
<code>\ifFB@xetex@punct</code> : New counter <code>\FB@nonchar</code> needed for non			