

A Babel language definition file for French

frenchb.dtx v3.1k, 2016/01/01

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	13
2.1 Initial setup	13
2.2 Punctuation	16
2.2.1 Punctuation with LuaTeX	17
2.2.2 Punctuation with XeTeX	24
2.2.3 Punctuation with standard (pdf)TeX	27
2.2.4 Punctuation switches common to all engines	29
2.3 Commands for French quotation marks	31
2.4 Date in French	35
2.5 Extra utilities	35
2.6 Formatting numbers	40
2.7 Caption names	41
2.8 Dots...	45
2.9 More checks about packages' loading order	46
2.10 Setup options: keyval stuff	47
2.11 French lists	59
2.12 French indentation of sections	63
2.13 Formatting footnotes	64
2.14 Clean up and exit	66
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.

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

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

Changes between version 2.0 and v3.1k 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.

`frenchb` takes account of babel’s *main language* defined as the *last* option at babel’s loading. When French is not babel’s main language, `frenchb` does not alter the general layout of the document (even in parts where French is the current language): the layout of lists, footnotes, indentation of first paragraphs of sections are not customised by `frenchb`.

When French is loaded as the last option of babel, `frenchb` makes the following changes to the global layout, *both in French and in all other languages*²:

1. the first paragraph of each section is indented (L^AT_EX only);
2. the default items in itemize environment are set to ‘—’ instead of ‘•’, and all vertical spacing and glue is deleted; it is possible to change ‘—’ to something else (‘-’ for instance) using `\frenchbsetup{}` (see section 1.2 p. 4);
3. vertical spacing in general L^AT_EX lists is shortened;
4. footnotes are displayed “à la française”.
5. the separator following the table or figure number in captions is printed as ‘-’ instead of ‘:’; for changing this see 1.2.2 p. 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 v3.1k and was last revised on 2016/01/01.

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

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 frenchb to make typesetting easier:

1. French quotation marks can be entered using the commands `\og` and `\fg` which work in L^AT_EX 2_ε and PlainT_EX, their appearance depending on what is available to draw them; even if you use L^AT_EX 2_ε and T1-encoding, you should refrain from entering them as `<<~French quotation~>>`: `\og` and `\fg` provide better horizontal spacing (controlled by `\FBguillspace`). If French quote characters are available on your keyboard, you can use them, to get proper spacing in L^AT_EX 2_ε see option `og=«`, `fg=»` p. 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 frenchb 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. frenchb has been designed to take advantage of the `xspace` package if present: adding `\usepackage{xspace}` in the preamble will force macros like `\fg`, `\ier`, `\ieme`, `\dots`, ..., to respect the spaces you type after them, for instance typing ‘`1\ier juin`’ will print ‘1^{er} juin’ (no need for a forced space after `1\ier`).

1.2 Customisation

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

1.2.1 `\frenchbsetup{options}`

`\frenchbsetup{ShowOptions}` prints all available options to the `.log` file, it is just meant as a remainder of the list of offered options. As usual with `keyval`

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

`StandardLists=true (false*)` forbids frenchb to customise any kind of list.

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

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

`FrenchFootnotes=false (true*)` reverts to the standard layout of footnotes. By default frenchb typesets leading numbers as ‘1. ’ instead of ‘¹’, but has no effect on footnotes numbered with symbols (as in the `\thanks` command). Two commands `\StandardFootnotes` and `\FrenchFootnotes` are available to change the layout of footnotes locally; `\StandardFootnotes` can help when some footnotes are numbered with letters (inside minipages for instance).

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

`FrenchSuperscripts=false (true)` ; then `\up=\textsuperscript`. (option added in version 2.1). Should only be made `false` to recompile 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 frenchb 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 frenchb in that area should be fine in most circumstances.

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

stick to the default option and use the provided `\NoAutoSpacing` command inside a group in case an unwanted space is added by frenchb (i.e. `{\NoAutoSpacing 10:55}`).

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

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

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

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

`OldFigTabCaptions=true (false)` is to be used when figures’ and tables’ captions must be typeset as with pre 3.0 versions of frenchb (with `\CaptionSeparator` in French and colon otherwise). Intended for standard \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 frenchb’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). frenchb’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 frenchb which characters are opening and closing French guillemets (they depend on the input encoding), then you can type either « guillemets » or «guillemets» (with or without spaces) to get properly typeset French quotes. This option works with LuaLaTeX and XeLaTeX; with pdfLaTeX it requires `inputenc` to be loaded with a proper encoding: 8-bits encoding (latin1, latin9, ansinew, applemac,...) or multi-byte encoding (utf8, utf8x).

Options’ order – Please remember that options are read in the order they appear in the `\frenchbsetup{}` command. Someone wishing that frenchb leaves the layout of lists and footnotes untouched but caring for indentation of first paragraph of sections should choose

`\frenchbsetup{StandardLayout,IndentFirst}` to get the expected layout. The reverse order `\frenchbsetup{IndentFirst,StandardLayout}` would lead to option `IndentFirst` being overwritten by `StandardLayout`.

1.2.2 Captions

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

When French is the main language, by default (see below) frenchb 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 frenchb 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 frenchb (using `\CaptionSeparator` in French and colon in other languages); this option only makes sense with the standard L^AT_EX classes `article`, `report` and `book`. 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 3.1?

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 frenchb's version number to 3.0a:

- babel 3.9 is required now to process frenchb.ldf, this change allows for cleaner definitions of dates and captions for the Unicode engines LuaTeX and XeTeX and also provides a simpler syntax for end-users, see section 1.2.2 p.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.
- frenchb no longer loads frenchb.cfg: customisation should definitely be done using \frenchbsetup{} options.
- Description lists labels are now indented; set \listindentFB=0pt to get the former layout.
- The last but not least change affects the (recent) LuaTeX-based engines, (this means version 0.76 as included in TL2013 and up): active characters are no longer used in French for 'high punctuation'. Functionalities and user interface are unchanged.

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

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

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

What's new in version 2.6?

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

frenchb is now compatible with the paralist package.

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

What's new in version 2.5?

The main change is that active characters are no longer used in French with (recent) XeTeX-based engines (they still are with TeX-based engines). All the

functionalities (automatic insertion of missing spaces before `;!?` or bare replacement of typed spaces with suitable unbreakable ones, tuning of the spaces width) remain available and the user interface is unchanged. The use of active characters is replaced by the `\XeTeXinterchartoks` mechanism (adapted from the `polyglossia` package).

A new command `\NoAutoSpacing` has been added. It should be used *inside a group* instead of `\shorthandoff{;!?}` whenever active characters or automatic spacing of French punctuation or quote characters conflict with other packages; it is designed to work with TeX-, LuaTeX- and XeTeX-based engines.

Bug corrections: `\frenchspacing` and `\nonfrenchspacing` are no longer messed up by `frenchb.ldf`.

What's new in version 2.4?

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

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

Better compatibility with the `enumitem` package.

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

What's new in version 2.3?

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

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

What's new in version 2.2?

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

What's new in version 2.1?

A new command `\fup` is provided to typeset better looking superscripts; it was designed using ideas from Jacques André, Thierry Bouche and René Fritz,

thanks to all of them! Former command `\up` is now defined as `\fup`, an option `FrenchSuperscripts=false` is provided for backward compatibility.

What's new in version 2.0?

Here is the list of all changes:

- Support for \LaTeX-2.09 and for $\text{\LaTeX2}_{\epsilon}$ in compatibility mode has been dropped. This version is meant for $\text{\LaTeX2}_{\epsilon}$ and Plain based formats (like `bplain`). $\text{\LaTeX2}_{\epsilon}$ formats based on `\text{\LaTeX}` are no longer supported either (plenty of good 8-bits fonts are available now, so T1 encoding should be preferred for typesetting in French). A warning is issued when OT1 encoding is in use at the `\begin{document}`.
- Customisation should now be handled only by command `\frenchbsetup{}`, `frenchb.cfg` (kept for compatibility) should no longer be used. See section 1.2 for the list of available options.
- Captions in figures and tables have changed in French: former abbreviations “Fig.” and “Tab.” have been replaced by full names “Figure” and “Table”. If this leads to formatting problems in captions, you can add the following two commands to your preamble (after loading `babel`) to get the former captions

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

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

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

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

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

2 The code

2.1 Initial setup

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

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

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

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

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

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

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

```

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

```

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

```

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

```

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

```

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

```

```
81 \bbl@tempa
```

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

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

```
82 \ifx\l@acadian\@undefined
83   \ifx\l@canadien\@undefined
84     \adddialect\l@acadian\l@french
85     \adddialect\l@canadien\l@french
86   \else
87     \adddialect\l@acadian\l@canadien
88   \fi
89 \else
90   \adddialect\l@canadien\l@acadian
91 \fi
92 \def\bbl@tempa{canadien}
93 \ifx\CurrentOption\bbl@tempa
94   \def\captionscanadien{\captionzacadian}
95   \def\datecanadien{\datezacadian}
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, `frenchb` handles French quotes together with ‘high punctuation’, a new conditional will be needed:

```

129 \newif\ifFBAutoSpaceGuill \FBAutoSpaceGuilltrue

```

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

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


```

130 \newif\ifFB@active@punct \FB@active@puncttrue
131 \newif\ifFB@luatex@punct
132 \ifBLaTeX
133   \ifnum\luatexversion>75
134     \FB@luatex@puncttrue\FB@active@punctfalse
135   \fi
136 \fi

```

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

```

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

```

\FBcolonspace According to the I.N. specifications, the ‘:’ requires an inter-word space before it, 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. 19). `\FBcolonskip` and `\FBthinskip` are also user customisable.

```

143 \newcommand*{\FBcolonspace}{\space}
144 \newcommand*{\FBthinspace}{\hskip .16667em \relax}
145 \newskip\FBcolonskip
146 \FBcolonskip=3.33pt plus 1.665pt minus 1.11pt \relax
147 \newskip\FBthinskip
148 \FBthinskip=1.66672pt \relax

```

2.2.1 Punctuation with LuaTeX

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

The following `\directlua` call ensures compatibility with LaTeX releases prior to 2015/10/01: the `\lcalleftbox` primitive⁴ introduced by Omega was prefixed with “`luatex`”, it should no longer be, see `ltnews23.tex` for details.

```

149 \ifB@luatex@punct
150   \directlua{tex.enableprimitives("", tex.extraprimatives("omega"))}

```

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

```

151 \begingroup\expandafter\expandafter\expandafter\endgroup
152 \expandafter\ifx\csname newluafunction\endcsname\relax

```

This code is for Plain or LaTeX versions prior to 2015/10/01.

```

153 \ifLaTeXe

```

⁴used by `\frquote`, see p. 34.

```

154     \AtEndOfPackage{%
155         \RequirePackage{luatexbase}%
156         \newluatexattribute\FB@addDPspace \FB@addDPspace=1 \relax
157         \newluatexattribute\FB@addGUIspace \FB@addGUIspace=0 \relax
158     }
159     \else
160         \input luatexbase.sty
161         \newluatexattribute\FB@addDPspace \FB@addDPspace=1 \relax
162         \newluatexattribute\FB@addGUIspace \FB@addGUIspace=0 \relax
163     \fi
164     \else

```

This code is for recent LaTeX versions (starting with 2015/10/01) or Plain when `l\luatex.tex` has been loaded before `babel`.

```

165     \newattribute\FB@addDPspace \FB@addDPspace=1 \relax
166     \newattribute\FB@addGUIspace \FB@addGUIspace=0 \relax
167     \fi
168     \ifLaTeXe
169         \PackageInfo{frenchb.ldf}{No need for active punctuation
170             characters\MessageBreak with this version
171             of LuaTeX!\MessageBreak reported}
172     \else
173         \fb@info{No need for active punctuation characters\\
174             with this version of LuaTeX!}
175     \fi
176 \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).

```

177 local FB_punct_thin =
178   {[string.byte("!")] = true,
179    [string.byte("?")] = true,
180    [string.byte(";")] = true}
181 local FB_punct_thick =
182   {[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 ‘»’.

```

183 local FB_punct_left =
184   {[string.byte("!")] = true,
185    [string.byte("?")] = true,
186    [string.byte(";")] = true,
187    [string.byte(":")] = true,
188    [0x14] = true,
189    [0xBB] = true}
190 local FB_punct_right =

```

```

191  {[0x13]                = true,
192  [0xAB]                  = true}

```

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

```

193 local FB_punct_null =
194  {[string.byte("!")] = true,
195   [string.byte("?")] = true,
196   [string.byte("[") = true,
197   [string.byte("(") = true,

```

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

```

198  [0xA0]                  = true,
199  [0x202F]                = true}
200 local FB_guil_null =
201  {[0xA0]                  = true,
202  [0x202F]                = true}

```

Local definitions for nodes:

```

203 local new_node          = node.new
204 local copy_node         = node.copy
205 local node_id           = node.id
206 local HLIST             = node_id("hlist")
207 local TEMP              = node_id("temp")
208 local KERN              = node_id("kern")
209 local GLUE              = node_id("glue")
210 local GSPEC             = node_id("glue_spec")
211 local GLYPH             = node_id("glyph")
212 local PENALTY           = node_id("penalty")
213 local nobreak           = new_node(PENALTY)
214 nobreak.penalty         = 10000
215 local insert_node_before = node.insert_before
216 local insert_node_after  = node.insert_after
217 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...

```

218 local thin10 = tex.skip['FBthinskip']
219 local thinwd = thin10.width/65536/3.33
220 local thinst = thin10.stretch/65536/1.665
221 local thinsh = thin10.shrink/65536/1.11
222 local coln10 = tex.skip['FBcolonskip']
223 local colnwd = coln10.width/65536/3.33
224 local colnst = coln10.stretch/65536/1.665
225 local colnsh = coln10.shrink/65536/1.11
226 local guil10 = tex.skip['FBguillskip']
227 local guilwd = guil10.width/65536/3.33
228 local guilst = guil10.stretch/65536/1.665
229 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):

```

230 local font_table = {}
231 local function new_glue_scaled (fid,width,stretch,shrink)
232   if fid > 0 then
233     local fp = font_table[fid]
234     if not fp then
235       font_table[fid] = font.getfont(fid).parameters
236       fp = font_table[fid]
237     end
238     local gl = new_node(GLUE,0)
239     local gl_spec = new_node(GSPEC)
240     gl_spec.width = width * fp.space
241     gl_spec.stretch = stretch * fp.space_stretch
242     gl_spec.shrink = shrink * fp.space_shrink
243     gl.spec = gl_spec
244     return gl
245   else
246     return nil
247   end
248 end

```

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

```

249 local addDPspace   = luatexbase.attributes['FB@addDPspace']
250 local addGUILspace = luatexbase.attributes['FB@addGUILspace']
251 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`).

```

252 local function french_punctuation (head)
253   for item in node.traverse_id(GLYPH, head) do
254     local lang = item.lang
255     local char = item.char
256     local fid  = item.font
257     local SIG  = has_attribute(item, addGUILspace)
258     if lang == FR and FB_punct_left[char] and fid > 0 then
259       local prev = item.prev
260       local prev_id, prev_subtype, prev_char
261       if prev then
262         prev_id = prev.id
263         prev_subtype = prev.subtype
264         if prev_id == GLYPH then
265           prev_char = prev.char
266         end
267       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.

```

268     local glue = prev_id == GLUE and prev_subtype == 0
269     local glue_wd
270     if glue then
271         glue_spec = prev.spec
272         glue_wd = glue_spec.width
273     end
274     local realglue = 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) preceeds the punctuation character, c) the punctuation character starts a paragraph or an `\hbox{}`.

```

275     if FB_punct_thin[char] or FB_punct_thick[char] then
276         local SBDP = has_attribute(item, addDPspace)
277         local auto = SBDP and SBDP > 0
278         if auto then
279             if (prev_char and FB_punct_null[prev_char]) or
280                 (glue and glue_wd <= 1) or
281                 (prev_id == HLIST and prev_subtype == 3) or
282                 (prev_id == TEMP) then
283                 auto = false
284             end
285         end
286         local fbglue
287         if FB_punct_thick[char] then
288             fbglue = new_glue_scaled(fid,colnwd,colnst,colnsh)
289         else
290             fbglue = new_glue_scaled(fid,thinwd,thinst,thinsh)
291         end
292         if realglue or auto then
293             if realglue then
294                 head = remove_node(head,prev,true)
295             end
296             insert_node_before(head, item, copy_node(nobreak))
297             insert_node_before(head, item, copy_node(fbglue))
298         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.

```

299         elseif SIG and SIG > 0 then
300             local addgl = (prev_char and not FB_guil_null[prev_char]) or
301                           (not prev_char and
302                             prev_id ~= TEMP and
303                             not (prev_id == HLIST and prev_subtype == 3)
304                           )

```

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

```

305         if glue and glue_wd <= 1 then
306             addgl = false
307         end
308         if addgl then
309             if glue then
310                 head = remove_node(head,prev,true)
311             end
312             local fbglue = new_glue_scaled(fid,guilwd,guilst,guilsh)
313             insert_node_before(head, item, copy_node(nobreak))
314             insert_node_before(head, item, copy_node(fbglue))
315         end
316     end
317 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*.

```

318     if lang == FR and FB_punct_right[char] and fid > 0
319                                     and SIG and SIG > 0 then
320         local next = item.next
321         local next_id, next_subtype, next_char, nextnext, kern_wd
322         if next then
323             next_id = next.id
324             next_subtype = next_subtype
325             if next_id == GLYPH then
326                 next_char = next.char

```

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

```

327         elseif next_id == KERN then
328             kern_wd = next.kern
329             if kern_wd == 0 then
330                 nextnext = next.next
331                 if nextnext then
332                     next = nextnext
333                     next_id = nextnext.id
334                     next_subtype = nextnext_subtype
335                     if next_id == GLYPH then
336                         next_char = nextnext.char

```

```

337         end
338     end
339 end
340 end
341 end
342 local glue = next_id == GLUE and next_subtype == 0
343 if glue then
344     glue_spec = next.spec
345     glue_wd = glue_spec.width
346 end
347 local addgl = (next_char and not FB_guil_null[next_char]) or
348 (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).

```

349     if glue and glue_wd == 0 then
350         addgl = false
351     end
352     if addgl then
353         if glue then
354             head = remove_node(head,next,true)
355         end
356         local fid = item.font
357         local fbglue = new_glue_scaled(fid,guilwd,guilst,guilsh)
358         insert_node_after(head, item, copy_node(fbglue))
359         insert_node_after(head, item, copy_node(nobreak))
360     end
361 end
362 end
363 return head
364 end
365 return french_punctuation

```

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

```

366 \ifFB@luatex@punct
367   \newcommand*{\FB@luatex@punct@french}{%
368     \ifx\shorthandoffORI\undefined
369       \let\shorthandonORI\shorthandon
370       \let\shorthandoffORI\shorthandoff
371     \fi
372     \def\shorthandoff##1{%
373       \ifx\PackageWarning\undefined
374         \fb@warning{\noexpand\shorthandoff{;:!?} is helpless with
375           LuaTeX,\, use \noexpand\NoAutoSpacing
376           *inside a group* instead.}%
377       \else
378         \PackageWarning{frenchb.ldf}{\protect\shorthandoff{;:!?} is
379           helpless with LuaTeX,\MessageBreak use \protect\NoAutoSpacing
380           \space *inside a group* instead;\MessageBreak reported}%

```

```

381     \fi}%
382     \def\shorthandon##1{%
383   }
384   \newcommand*\FB@luatex@punct@nonfrench{%
385     \ifx\shorthandoffORI\@undefined
386     \else
387       \let\shorthandon\shorthandonORI
388       \let\shorthandoff\shorthandoffORI
389     \fi
390   }
391   \FB@addto{extras}{\FB@luatex@punct@french}
392   \FB@addto{noextras}{\FB@luatex@punct@nonfrench}

```

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

```

393   \def\activate@luatexpunct{%
394     \directlua{%
395       FR = \the\l@french
396       local path = kpse.find_file("frenchb.lua", "lua")
397       if path then
398         local f = dofile(path)
399         luatexbase.add_to_callback("kerning",
400           f, "frenchb.french_punctuation")
401       else
402         texio.write_nl('')
403         texio.write_nl('*****')
404         texio.write_nl('Error: frenchb.lua not found.')
405         texio.write_nl('*****')
406         texio.write_nl('')
407       end
408     }%
409   }
410 \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 255 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.

```

411 \newcount\FB@interchartokenstateORI
412 \ifFB@xetex@punct
413   \ifLaTeXe
414     \PackageInfo{frenchb.ldf}{No need for active punctuation characters%
415       \MessageBreak with this version of XeTeX!%
416       \MessageBreak reported}
417   \else
418     \fb@info{No need for active punctuation characters\\
419       with this version of XeTeX!}
420   \fi

```

Six new character classes are defined for frenchb.

```

421 \newXeTeXintercharclass\FB@punctthick
422 \newXeTeXintercharclass\FB@punctthin
423 \newXeTeXintercharclass\FB@punctnul
424 \newXeTeXintercharclass\FB@guilo
425 \newXeTeXintercharclass\FB@guilf
426 \newXeTeXintercharclass\FB@guilnul

```

We define a command to store the `\XeTeXcharclass` values which will be modified for French (as a comma separated list) and a command to retrieve them.

```

427 \def\FB@charclassesORI{}
428 \def\empty{}
429 \def\FB@parse#1,#2\endparse{\def\FB@class{#1}%
430   \def\FB@charclassesORI{#2}}%

```

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

```

431 \newcommand*{\FB@xetex@punct@french}{%

```

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

```

432   \ifx\FB@charclassesORI\empty
433     \FB@interchartokenstateORI=\XeTeXinterchartokenstate
434     \bbl@for\FB@char
435       {\:,\;,\!,\?, "AB,"BB,%
436         \'(,\[,\'{,\,,\'.,\'-,\'\\),\'\\),\'},%
437         \'%, "22,"27,"60,"2019,"A0,"202F}%
438       {\edef\FB@charclassesORI{\FB@charclassesORI%
439         \theXeTeXcharclass\FB@char,}}%
440     \let\shorthandonORI\shorthandon
441     \let\shorthandoffORI\shorthandoff
442   \fi

```

Set the classes and interactions between classes.

```

443   \XeTeXinterchartokenstate=1
444   \XeTeXcharclass \: = \FB@punctthick

```

```

445 \XeTeXinterchartoks \z@ \FB@punctthick = {%
446     \ifhmode\FDP@colonspace\fi}%
447 \XeTeXinterchartoks \FB@guilf \FB@punctthick = {%
448     \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=255` isn’t specific to glue tokens (this class includes box and math boundaries f.i.), so the `\else` part cannot be omitted.

```

449 \XeTeXinterchartoks 255 \FB@punctthick = {%
450     \ifhmode
451         \ifdim\lastskip>1sp
452             \unskip\penalty\@M\FBcolonspace
453         \else
454             \FDP@colonspace
455         \fi
456     \fi}%
457 \bbl@for\FB@char
458     {\‘\;,\‘!\,\‘?}%
459     {\XeTeXcharclass\FB@char=\FB@punctthin}%
460 \XeTeXinterchartoks \z@ \FB@punctthin = {%
461     \ifhmode\FDP@thinspace\fi}%
462 \XeTeXinterchartoks \FB@guilf \FB@punctthin = {%
463     \FDP@thinspace}%
464 \XeTeXinterchartoks 255 \FB@punctthin = {%
465     \ifhmode
466         \ifdim\lastskip>1sp
467             \unskip\penalty\@M\FBthinspace
468         \else
469             \FDP@thinspace
470         \fi
471     \fi}%
472 \XeTeXinterchartoks \FB@guilo \z@ = {%
473     \ifFBAutoSpaceGuill\FBguillspace\fi}%
474 \XeTeXinterchartoks \FB@guilo 255 = {%
475     \ifFBAutoSpaceGuill\FBguillspace\ignorespaces\fi}%
476 \XeTeXinterchartoks \z@ \FB@guilf = {%
477     \ifFBAutoSpaceGuill\FBguillspace\fi}%
478 \XeTeXinterchartoks \FB@punctthin \FB@guilf = {%
479     \ifFBAutoSpaceGuill\FBguillspace\fi}%
480 \XeTeXinterchartoks 255 \FB@guilf = {%
481     \ifFBAutoSpaceGuill\unskip\FBguillspace\fi}%

```

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

```

482 \bbl@for\FB@char
483     {\‘\[, \‘\[, "A0, "202F}%
484     {\XeTeXcharclass\FB@char=\FB@punctnul}%

```

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

```

485 \bbl@for\FB@char
486   {'\{','\,, '\., '\-, '\), '\}, '\%, "22, "27, "60, "2019}%
487   {\XeTeXcharclass\FB@char=\z@}%

```

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

```

488 \def\shorthandoff##1{%
489   \ifx\PackageWarning\@undefined
490     \fb@warning{\noexpand\shorthandoff{;:!?} is helpless with
491       XeTeX,\, use \noexpand\NoAutoSpacing
492       *inside a group* instead.}%
493   \else
494     \PackageWarning{frenchb.ldf}{\protect\shorthandoff{;:!?} is
495       helpless with XeTeX,\MessageBreak use \protect\NoAutoSpacing
496       \space *inside a group* instead;\MessageBreak reported}%
497   \fi}%
498 \def\shorthandon##1{%
499   }

```

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

```

500 \newcommand*\FB@xetex@punct@nonfrench{%
501   \ifx\FB@charclassesORI\empty
502   \else
503     \bbl@for\FB@char
504       {'\:, '\;, '\!, '\?, "AB, "BB, %
505       '\(, '\[, '\{, '\,, '\., '\-, '\), '\], '\}, %
506       '\%, "22, "27, "60, "2019, "A0, "202F}%
507     {\expandafter\FB@parse\FB@charclassesORI\endparse
508     \XeTeXcharclass\FB@char=\FB@class}%
509   \def\FB@charclassesORI{%
510     \XeTeXinterchartokenstate=\FB@interchartokenstateORI
511     \let\shorthandon\shorthandonORI
512     \let\shorthandoff\shorthandoffORI
513   \fi
514   }
515   \FB@addto{extras}{\FB@xetex@punct@french}
516   \FB@addto{noextras}{\FB@xetex@punct@nonfrench}

```

End of specific code for punctuation with modern XeTeX engines.

```
517 \fi
```

2.2.3 Punctuation with standard (pdf)TeX

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

```

518 \ifFB@active@punct
519   \initiate@active@char{:}%
520   \initiate@active@char{;}%

```

```

521 \initiate@active@char{!}%
522 \initiate@active@char{?}%

```

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

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

```

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

```

Now we can insert a ; character.

```

531     \string;}

```

The next three definitions are very similar.

```

532 \declare@shorthand{french}{!}{%
533     \ifhmode
534         \ifdim\lastskip>1sp
535             \unskip\penalty\@M\FBthinspace
536         \else
537             \FDP@thinspace
538         \fi
539     \fi
540     \string!}
541 \declare@shorthand{french}{?}{%
542     \ifhmode
543         \ifdim\lastskip>1sp
544             \unskip\penalty\@M\FBthinspace
545         \else
546             \FDP@thinspace
547         \fi
548     \fi
549     \string?}
550 \declare@shorthand{french}{:}{%
551     \ifhmode
552         \ifdim\lastskip>1sp
553             \unskip\penalty\@M\FBcolonspace
554         \else
555             \FDP@colonspace
556         \fi
557     \fi
558     \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.

```
559 \declare@shorthand{system}{:}{\string:}
560 \declare@shorthand{system}{!}{\string!}
561 \declare@shorthand{system}{?}{\string?}
562 \declare@shorthand{system}{;}{\string;}
563 %}
```

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

```
564 \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.

```
565 \bbl@activate{:}\bbl@activate{;}%
566 \bbl@activate{!}\bbl@activate{?}%
567 }
568 \FB@addto{noextras}{%
569 \bbl@deactivate{:}\bbl@deactivate{;}%
570 \bbl@deactivate{!}\bbl@deactivate{?}%
571 }
572 \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.

```
573 \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).

```
574 \def\autospace@beforeFDP{%
575     \ifFB@luatex@punct\FB@addDPspace=1 \fi
576     \def\FDP@thinspace{\penalty\@M\FBthinspace}%
577     \def\FDP@colonspace{\penalty\@M\FBcolonspace}%
578 \def\noautospace@beforeFDP{%
579     \ifFB@luatex@punct\FB@addDPspace=0 \fi
580     \let\FDP@thinspace\@empty
581     \let\FDP@colonspace\@empty}
582 \ifLaTeXe
583     \def\AutoSpaceBeforeFDP{\autospace@beforeFDP
584                             \FBAutoSpacePunctuationtrue}
585     \def\NoAutoSpaceBeforeFDP{\noautospace@beforeFDP
586                             \FBAutoSpacePunctuationfalse}
```

```

587 \AtEndOfPackage{\AutoSpaceBeforeFDP}
588 \else
589 \let\AutoSpaceBeforeFDP\autospace@beforeFDP
590 \let\NoAutoSpaceBeforeFDP\noautospace@beforeFDP
591 \AutoSpaceBeforeFDP
592 \fi

```

\rmfamilyFB In \LaTeX_ϵ **\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.

```

593 \ifLaTeXe
594 \DeclareRobustCommand\ttfamilyFB{%
595 \FBAutoSpaceGuillfalse
596 \ifFB@luatex@punct\FB@addGUILspace=0 \fi
597 \noautospace@beforeFDP\ttfamilyORI}%
598 \DeclareRobustCommand\rmfamilyFB{%
599 \FBAutoSpaceGuilltrue
600 \ifFB@luatex@punct\FB@addGUILspace=1 \fi
601 \ifFBAutoSpacePunctuation
602 \autospace@beforeFDP
603 \else
604 \noautospace@beforeFDP
605 \fi
606 \rmfamilyORI}%
607 \DeclareRobustCommand\sffamilyFB{%
608 \FBAutoSpaceGuilltrue
609 \ifFB@luatex@punct\FB@addGUILspace=1 \fi
610 \ifFBAutoSpacePunctuation
611 \autospace@beforeFDP
612 \else
613 \noautospace@beforeFDP
614 \fi
615 \sffamilyORI}%
616 \fi

```

\NoAutoSpacing The following command will switch off active punctuation characters (if any) and disable automatic spacing for French quote characters. It is engine independent (works for TeX, LuaTeX and XeTeX based engines) and is meant to be used inside a group.

```

617 \newcommand*{\NoAutoSpacing}{\FBAutoSpaceGuillfalse

```

```

618 \ifFB@active@punct\shorthandoff{;:!?}\fi
619 \ifFB@xetex@punct\XeTeXinterchartokenstate=0 \fi
620 \ifFB@luatex@punct\FB@addDPspace=0 \FB@addGUILspace=0 \fi
621 }

```

2.3 Commands for French quotation marks

\og The top macros for quotation marks will be called **\og** (“ouvrez guillemets”) and **\fg** (“fermez guillemets”). Another option for typesetting quotes in 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. The default definition of **\og** and **\fg** will be set later (for English) by **\bbl@nonfrenchguillemets**.

```

622 \newcommand*{\og}{\@empty}
623 \newcommand*{\fg}{\@empty}

```

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

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

```

624 \ifLaTeXe
625 \else
626 \ifFBunicode
627 \def\guillemotleft{\char"00AB}
628 \def\guillemotright{\char"00BB}
629 \def\textquotedblleft{\char"201C}
630 \def\textquotedblright{\char"201D}
631 \else
632 \def\guillemotleft{\leavevmode\raise0.25ex
633 \hbox{$\scriptscriptstyle\ll$}}
634 \def\guillemotright{\raise0.25ex
635 \hbox{$\scriptscriptstyle\gg$}}
636 \def\textquotedblleft{' '}
637 \def\textquotedblright{' '}
638 \fi
639 \let\xspace\relax
640 \fi

```

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

a space after `\fg`, otherwise `\xspace` will be defined as `\relax` (done at the end of this file).

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

```
641 \newskip\FBguillskip
642 \FBguillskip=2.664pt plus 0.500pt minus 0.888pt \relax
643 \newcommand*\FBguillspace{\penalty\@M\hskip.8\fontdimen2\font
644                                     plus.3\fontdimen3\font
645                                     minus.8\fontdimen4\font}
```

`\FBguillspace` is not used with LuaTeX.

```
646 \ifFB@luatex@punct
647   \DeclareRobustCommand*\FB@og{\leavevmode
648       \bgroup\FB@addGUILLspace=1 \guillemotleft\egroup}
649   \DeclareRobustCommand*\FB@fg{\ifdim\lastskip>\z@\unskip\fi
650       \bgroup\FB@addGUILLspace=1 \guillemotright\egroup\xspace}
651 \fi
```

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

```
652 \ifFB@xetex@punct
653   \DeclareRobustCommand*\FB@og{\leavevmode
654       \bgroup\FBAutoSpaceGuillfalse\guillemotleft\egroup
655       \FBguillspace}
656   \DeclareRobustCommand*\FB@fg{\ifdim\lastskip>\z@\unskip\fi
657       \FBguillspace
658       \bgroup\FBAutoSpaceGuillfalse\guillemotright\egroup\xspace}
659 \fi
660 \ifFB@active@punct
661   \DeclareRobustCommand*\FB@og{\leavevmode
662       \guillemotleft
663       \FBguillspace}
664   \DeclareRobustCommand*\FB@fg{\ifdim\lastskip>\z@\unskip\fi
665       \FBguillspace
666       \guillemotright\xspace}
667 \fi
```

The top level definitions for French quotation marks are switched on and off through the `\extrasfrench` `\noextrasfrench` mechanism. Outside French, `\og` and `\fg` will typeset standard English opening and closing double quotes.

```
668 \ifLaTeXe
669   \def\bbf@frenchguillemets{\renewcommand*\og{\FB@og}%
670       \renewcommand*\fg{\FB@fg}}
671   \renewcommand*\og{\textquotedblleft}
672   \renewcommand*\fg{\ifdim\lastskip>\z@\unskip\fi \textquotedblright}
673 \else
674   \def\bbf@frenchguillemets{\let\og\FB@og
675       \let\fg\FB@fg}
676   \def\og{\textquotedblleft}
677   \def\fg{\ifdim\lastskip>\z@\unskip\fi \textquotedblright}
678 \fi
679 \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...

```
680 \newcommand*{\ogi}{\FB@og}
681 \newcommand*{\fgi}{\FB@fg}
682 \newcommand*{\ogii}{\textquotedblleft}
683 \newcommand*{\fgii}{\textquotedblright}

and the needed technical stuff to handle options:

684 \newcount\FBguill@level
685 \newtoks\FB@everypar
686 \newif\ifFBcloseguill \FBcloseguilltrue
687 \newif\ifFBInnerGuillSingle
688 \def\FBguillopen{\bgroup\NoAutoSpacing\guillemotleft\egroup}
689 \def\FBguillclose{\bgroup\NoAutoSpacing\guillemotright\egroup}
690 \let\FBguillnone\relax
691 \let\FBeveryparguill\FBguillopen
692 \ifFB@luatex@punct
693   \let\FBeverylanguill\FBguillopen
694 \else
695   \let\FBeverylanguill\FBguillnone
696 \fi
```

The main command `\frquote` accepts (in \LaTeX 2_ϵ 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.

```
697 \ifLaTeXe
698   \DeclareRobustCommand\frquote{%
699     \@ifstar{\FBcloseguillfalse\fr@quote}%
700     {\FBcloseguilltrue\fr@quote}}
701 \else
702   \newcommand\frquote[1]{\fr@quote{#1}}
703 \fi
```

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

```
704 \newcommand{\fr@quote}[1]{%
705   \leavevmode
706   \advance\FBguill@level by \@ne
```

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

```
707 \def\FB@quotespace{\kern.8\fontdimen2\font}%
708 \ifcase\FBguill@level
709 \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:

```
710   \FB@everypar=\everypar
711   \ifx\FBeveryparguill\relax
712   \else
713     \def\FBeverypar@quote{\FBeveryparguill\FB@quotespace}%
714     \everypar=\expandafter{\the\everypar \FBeverypar@quote}%

```

```

715 \fi
716 \ogi #1\fgi
717 \or

```

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

```

718 \ifx\FBverylineguill\FBguillopen
719 \localleftbox{\guillemotleft\FB@quotespace}%
720 \let\FBeverypar@quote\relax
721 \ogi #1\ifFBcloseguill\fgi\fi
722 \else
723 \ifx\FBverylineguill\FBguillclose
724 \localleftbox{\guillemotright\FB@quotespace}%
725 \let\FBeverypar@quote\relax
726 \ogi #1\ifFBcloseguill\fgi\fi
727 \else

```

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

```

728 \let\FBeverypar@quote\relax
729 \ifFBInnerGuillSingle
730 \def\ogii{\leavevmode
731 \guilsinglleft\FBguillspace}%
732 \def\fgii{\ifdim\lastskip>\z@\unskip\fi
733 \FBguillspace\guilsinglright}%
734 \ifx\FBveryparguill\FBguillopen
735 \def\FBeverypar@quote{\guilsinglleft\FB@quotespace}%
736 \fi
737 \ifx\FBveryparguill\FBguillclose
738 \def\FBeverypar@quote{\guilsinglright\FB@quotespace}%
739 \fi
740 \fi
741 \ogii #1\ifFBcloseguill \fgii \fi
742 \fi
743 \fi
744 \else

```

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

```

745 \ifx\PackageWarning\@undefined
746 \fb@warning{\noexpand\frquote\space accepts no more than
747 two levels.\\ Quotation not printed.}%
748 \else
749 \PackageWarning{frenchb.ldb}{%
750 \protect\frquote\space accepts no more than two levels
751 \MessageBreak Quotation not printed. Reported}
752 \fi
753 \fi

```

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

```

754 \advance\FBguill@level by \m@ne
755 \ifx\FBverylineguill\FBguillnone\else\localleftbox{}\fi

```

```

756 \ifx\FBeveryparguill\relax\else\everypar=\FB@everypar\fi
757 }

```

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.

```

758 \ifLaTeXe
759 \def\BabelLanguages{french,acadian}
760 \StartBabelCommands*\BabelLanguages}{date}
761 [unicode, fontenc=EU1 EU2, charset=utf8]
762 \SetString\monthiiname{février}
763 \SetString\monthviiiname{août}
764 \SetString\monthxiiname{décembre}
765 \StartBabelCommands*\BabelLanguages}{date}
766 \SetStringLoop{month#1name}{%
767 janvier,f\'evrier,mars,avril,mai,juin,juillet,%
768 ao\^ut,septembre,octobre,novembre,d\'ecembre}
769 \SetString\today{{\number\day}\ifnum1=\day {\ier}\fi\space
770 \csname month\romannumeral\month name\endcsname \space
771 \number\year
772 }
773 \EndBabelCommands
774 \else
775 \ifFBunicode
776 \@namedef{date\CurrentOption}{%
777 \def\today{{\number\day}\ifnum1=\day {\ier}\fi \space
778 \ifcase\month
779 \or janvier\or février\or mars\or avril\or mai\or
780 juin\or juillet\or août\or septembre\or
781 octobre\or novembre\or décembre\fi
782 \space \number\year}}
783 \else
784 \@namedef{date\CurrentOption}{%
785 \def\today{{\number\day}\ifnum1=\day {\ier}\fi \space
786 \ifcase\month
787 \or janvier\or f\'evrier\or mars\or avril\or mai\or
788 juin\or juillet\or ao\^ut\or septembre\or
789 octobre\or novembre\or d\'ecembre\fi
790 \space \number\year}}
791 \fi
792 \fi

```

2.5 Extra utilities

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

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

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

```
793 \newif\ifFB@poorman
794 \newdimen\FB@Mht
795 \ifLaTeXe
796 \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{}`.

```
797 \newcommand*\FBsupR{-0.12}
798 \newcommand*\FBsupS{0.65}
799 \newcommand*\FB@lc[1]{\MakeLowercase{#1}}
800 \DeclareRobustCommand*\FB@up@fake[1]{%
801   \settoheight{\FB@Mht}{M}%
802   \addtolength{\FB@Mht}{\FBsupR \FB@Mht}%
803   \addtolength{\FB@Mht}{-\FBsupS ex}%
804   \raisebox{\FB@Mht}{\scalefont{\FBsupS}{\FB@lc{#1}}}%
805 }
```

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.

```
806 \def\FB@split#1#2#3#4\@nil{\def\FB@firstthree{#1#2#3}%
807                               \def\FB@suffix{#4}}
808 \def\FB@x{x}
809 \def\FB@j{j}
```

```

810 \DeclareRobustCommand*\FB@up}[1]{%
811 \bgroup \FB@poormantrue
812 \expandafter\FB@split\fbfamily\@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.

```

813 \edef\reserved@a{\lowercase{%
814 \noexpand\IfFileExists{\fbencoding\FB@firstthree -sup.fd}}}%
815 \reserved@a
816 {\ifx\FB@suffix\FB@x \FB@poormanfalse\fi
817 \ifx\FB@suffix\FB@j \FB@poormanfalse\fi
818 \ifFB@poorman \FB@up@fake{#1}%
819 \else \FB@up@real{#1}%
820 \fi}%
821 {\FB@up@fake{#1}}}%
822 \egroup}

```

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

```

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

```

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

```

825 \DeclareRobustCommand*\fup}[1]{%
826 \ifx\realsuperscript\undefined
827 \FB@up{#1}%
828 \else
829 \bgroup\let\fakesuperscript\FB@up@fake
830 \realsuperscript{\FB@lc{#1}}\egroup
831 \fi}

```

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

```

832 \providecommand*\up{\relax}

```

Poor man's definition of \up for Plain.

```

833 \else
834 \providecommand*\up}[1]{\leavevmode\raiselex\hbox{\sevenrm #1}}
835 \fi

```

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

```

\ier 836 \def\ieme{\up{e}\xspace}
\iere 837 \def\iemes{\up{es}\xspace}
\iemes 838 \def\ier{\up{er}\xspace}
\iers 839 \def\iers{\up{ers}\xspace}
\ieres 840 \def\iere{\up{re}\xspace}
      841 \def\ieres{\up{res}\xspace}

```

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

```

\no 842 \newcommand*{\FrenchEnumerate}[1]{%
\nos 843      #1\up{o}\kern+.3em}
\nos 844 \newcommand*{\FrenchPopularEnumerate}[1]{%
\primo 845      #1\up{o})\kern+.3em}
\fprimo) Typing \primo should result in ‘1°’,
      846 \def\primo{\FrenchEnumerate1}
      847 \def\secundo{\FrenchEnumerate2}
      848 \def\tertio{\FrenchEnumerate3}
      849 \def\quarto{\FrenchEnumerate4}
      while typing \fprimo gives ‘1°’.
      850 \def\fprimo{\FrenchPopularEnumerate1}
      851 \def\fsecundo{\FrenchPopularEnumerate2}
      852 \def\ftertio{\FrenchPopularEnumerate3}
      853 \def\fquarto{\FrenchPopularEnumerate4}

      Let’s provide four macros for the common abbreviations of “Numéro”.
      854 \DeclareRobustCommand*\No{\N\up{o}\kern+.2em}
      855 \DeclareRobustCommand*\no{\n\up{o}\kern+.2em}
      856 \DeclareRobustCommand*\Nos{\N\up{os}\kern+.2em}
      857 \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 frenchb: a `\kern0pt` is used instead of `\hbox` because `\hbox` would break microtype’s font expansion; as a (positive?) side effect, composed names (such as Dupont-Durand) can now be hyphenated on explicit hyphens. Usage: Jean~`\bsc{Duchemin}`.

```

858 \DeclareRobustCommand*\bsc[1]{\leavevmode\begin{group}\kern0pt
859      \scshape #1\endgroup}
860 \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.

```

861 \ifFBunicode
862   \newcommand*\at{{\char"0040}}
863   \newcommand*\circonflexe{{\char"005E}}
864   \newcommand*\tild{{\char"007E}}
865   \newcommand*\boi{\textbackslash}
866   \newcommand*\degre{{\char"00B0}}
867 \else
868   \ifLaTeXe
869     \DeclareTextSymbol{\at}{T1}{64}
870     \DeclareTextSymbol{\circonflexe}{T1}{94}
871     \DeclareTextSymbol{\tild}{T1}{126}
872     \DeclareTextSymbolDefault{\at}{T1}
873     \DeclareTextSymbolDefault{\circonflexe}{T1}

```

```

874 \DeclareTextSymbolDefault{\tild}{T1}
875 \DeclareRobustCommand*\boi{\textbackslash}
876 \DeclareRobustCommand*\degree{\r{}}
877 \else
878 \def\T@one{T1}
879 \ifx\fontencoding\T@one
880 \newcommand*\degree{{\char6}}
881 \else
882 \newcommand*\degree{{\char23}}
883 \fi
884 \newcommand*\at{{\char64}}
885 \newcommand*\circonflexe{{\char94}}
886 \newcommand*\tild{{\char126}}
887 \newcommand*\boi{{\backslash}}
888 \fi
889 \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.

```

890 \ifLaTeXe
891 \newcommand*\degrees{\degree}
892 \ifFBunicode
893 \DeclareRobustCommand*\degrees{\degree}
894 \else
895 \def\Warning@degree@TSone{%
896 \PackageWarning{frenchb.ldf}{%
897 Degrees would look better in TS1-encoding:%
898 \MessageBreak add \protect
899 \usepackage{textcomp} to the preamble.%
900 \MessageBreak Degrees used}}
901 \AtBeginDocument{\ifx\DeclareEncodingSubset\@undefined
902 \DeclareRobustCommand*\degrees{%
903 \leavevmode\hbox to 0.3em{\hss\degree\hss}%
904 \Warning@degree@TSone
905 \global\let\Warning@degree@TSone\relax}%
906 \else
907 \DeclareRobustCommand*\degrees{%
908 \hbox{\UseTextSymbol{TS1}{\textdegree}}}%
909 \fi
910 }
911 \fi
912 \else
913 \newcommand*\degrees{%
914 \leavevmode\hbox to 0.3em{\hss\degree\hss}}

```

915 \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.

```
916 \newcount\std@mcc
917 \newcount\dec@mcc
918 \std@mcc=\mathcode'\,
919 \dec@mcc=\std@mcc
920 \@tempcnta=\std@mcc
921 \divide\@tempcnta by "1000
922 \multiply\@tempcnta by "1000
923 \advance\dec@mcc by -\@tempcnta
924 \newcommand*\DecimalMathComma{\iflanguage{french}%
925                               {\mathcode'\,=\dec@mcc}}}%
926   \FB@addto{extras}{\mathcode'\,=\dec@mcc}%
927 }
928 \newcommand*\StandardMathComma{\mathcode'\,=\std@mcc
929   \FB@addto{extras}{\mathcode'\,=\std@mcc}%
930 }
931 \FB@addto{noextras}{\mathcode'\,=\std@mcc}
```

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

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

```
932 \newcommand*\nombre[1]{\iflanguage{french}%
933                          {\fb@warning{*** \noexpand\nombre
no longer formats numbers\string! ***}}}
```

The next definitions only make sense for L^AT_EX 2_ε. For Plain based formats, let's activate LuaT_EX punctuation if necessary, then cleanup and exit. Temporary fix: `\l@french` is not properly set by babel 3.9h with Plain LuaT_EX format.

```
934 \let\FBstop@here\relax
935 \def\FBclean@on@exit{\let\ifLaTeXe\undefined
936                     \let\LaTeXtrue\undefined
937                     \let\LaTeXfalse\undefined}
938 \ifx\magnification\@undefined
939 \else
940   \def\FBstop@here{\ifFB@luatex@punct
941                   \activate@luatex@punct
942                   \fi
943   \FBclean@on@exit}
```



```

944 \ldf@quit\CurrentOption\endinput}
945 \fi
946 \FBstop@here

```

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

```

947 \renewcommand*{\nombre}[1]{\Warning@nombre{#1}}
948 \newcommand*{\Warning@nombre}[1]{%
949   \ifdefined\numprint
950     \numprint{#1}%
951   \else
952     \PackageWarning{frenchb.ldf}{%
953       \protect\nombre\space now relies on package numprint.sty,%
954       \MessageBreak add \protect
955       \usepackage[autolanguage]{numprint},\MessageBreak
956       see file numprint.pdf for more options.\MessageBreak
957       \protect\nombre\space called}%
958     \global\let\Warning@nombre\relax
959     {#1}%
960   \fi
961 }

```

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}`).

```

962 \ifx\FBfigtabshape\undefined \let\FBfigtabshape\scshape \fi

```

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

```

963 \StartBabelCommands*{\BabelLanguages}{captions}
964   [unicode, fontenc=EU1 EU2, charset=utf8]
965   \SetString{\refname}{Références}
966   \SetString{\abstractname}{Résumé}
967   \SetString{\prefacename}{Préface}
968   \SetString{\contentsname}{Table des matières}
969   \SetString{\ccname}{Copie à }
970   \SetString{\proofname}{Démonstration}
971   \SetStringLoop{ordinal#1}{%
972     Première,Deuxième,Troisième,Quatrième,Cinquième,%
973     Sixième,Septième,Huitième,Neuvième,Dixième,Onzième,%
974     Douzième,Treizième,Quatorzième,Quinzième,Seizième,%
975     Dix-septième,Dix-huitième,Dix-neuvième,Vingtième}
976 \StartBabelCommands*{\BabelLanguages}{captions}

```

```

977 \SetString{\refname}{R\’ef\’erences}
978 \SetString{\abstractname}{R\’esum\’e}
979 \SetString{\bibname}{Bibliographie}
980 \SetString{\prefacename}{Pr\’eface}
981 \SetString{\chaptername}{Chapitre}
982 \SetString{\appendixname}{Annexe}
983 \SetString{\contentsname}{Table des mati\’eres}
984 \SetString{\listfigurename}{Table des figures}
985 \SetString{\listtablename}{Liste des tableaux}
986 \SetString{\indexname}{Index}
987 \SetString{\figurename}{{\FBfigtabshape Figure}}
988 \SetString{\tablename}{{\FBfigtabshape Table}}
989 \SetString{\pagename}{page}
990 \SetString{\seename}{voir}
991 \SetString{\alsoname}{voir aussi}
992 \SetString{\enclname}{P.~J. }
993 \SetString{\ccname}{Copie \’a }
994 \SetString{\headtoname}{}
995 \SetString{\proofname}{D\’emonstration}
996 \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`.

```

997 \SetStringLoop{ordinal#1}{%
998     Premi\’ere,Deuxi\’eme,Troisi\’eme,Quatri\’eme,Cinqui\’eme,%
999     Sixi\’eme,Septi\’eme,Huiti\’eme,Neuvi\’eme,Dixi\’eme,Onzi\’eme,%
1000     Douzi\’eme,Treizi\’eme,Quatorzi\’eme,Quinzi\’eme,Seizi\’eme,%
1001     Dix-septi\’eme,Dix-huiti\’eme,Dix-neuvi\’eme,Vingti\’eme}
1002 \AfterBabelCommands{%
1003     \DeclareRobustCommand*{\FB@emptypart}{{\def\thepart{}}}%
1004     \DeclareRobustCommand*{\FB@partname}{%
1005         \ifFBPartNameFull
1006             \csname ordinal\romannumeral\value{part}\endcsname\space
1007             partie\FB@emptypart
1008         \else
1009             Partie%
1010         \fi}%
1011     }
1012 \SetString{\partname}{{\FB@partname}}
1013 \EndBabelCommands

```

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

```

1014 \AtEndOfPackage{%
1015     \ifFB@koma
1016         \ifdefined\partformat
1017             \FB@addto{captions}{%
1018                 \ifFBPartNameFull

```

```

1019             \babel@save\partformat
1020             \renewcommand*{\partformat}{\partname}%
1021             \fi}%
1022     \fi
1023 \fi
1024 }

```

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

\CaptionSeparator Let's consider now captions in figures and tables. In French, captions in figures and tables should never be printed as 'Figure 1:' which is the default in standard $\text{\LaTeX} 2_{\epsilon}$ classes; the ':' is made active too late, no space is added before it. With LuaLaTeX and XeLaTeX, this glitch doesn't occur, you get 'Figure 1 :' which is correct in French. With pdfLaTeX frenchb provides the following workaround.

The standard definition of `\@makecaption` (e.g., the one provided in `article.cls`, `report.cls`, `book.cls` which is frozen for $\text{\LaTeX} 2_{\epsilon}$ 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, frenchb just adds a hook called `\FBCaption@Separator` to `\@makecaption`; `\FBCaption@Separator` defaults to ':' as in the standard `\@makecaption` and will be changed to ':' in French 'AtBeginDocument'; it can be also set to `\CaptionSeparator` ('-') using [CustomiseFigTabCaptions](#).

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

```

1025 \bgroup
1026 \catcode':=12 \catcode'>=12 \relax
1027 \long\gdef\STD@makecaption#1#2{%
1028   \vskip\abovecaptionskip
1029   \sbox\@tempboxa{#1: #2}%
1030   \ifdim \wd\@tempboxa >\hsize
1031     #1: #2\par
1032   \else
1033     \global \@minipagefalse
1034     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
1035   \fi
1036   \vskip\belowcaptionskip}
1037 \egroup

```

The caption and floatrow packages are compatible with frenchb if they are loaded after babel (a warning is printed in the .log file when they are loaded too early).

No warning is issued for SMF and AMS classes as their layout of captions is compatible with French typographic standards.

With `memoir` and `koma-script` classes, frenchb customises `\captiondelim` or `\captionformat` in French (unless option [CustomiseFigTabCaptions](#) is set to `false`) and issues no warning.

When `\@makecaption` has been changed by another class or package, a warning is printed in the .log file.

```

1038 \newif\if@FBwarning@capsep
1039 \@FBwarning@capseptrue
1040 \newcommand{\FBWarning}[2]{\PackageWarning{#1}{#2}}
1041 \newcommand*\CaptionSeparator{\space\textendash\space}
1042 \def\FBCaption@Separator{: }
1043 \long\def\FB@makecaption#1#2{%
1044   \vskip\abovecaptionskip
1045   \sbox\@tempboxa{#1\FBCaption@Separator #2}%
1046   \ifdim \wd\@tempboxa >\hsize
1047     #1\FBCaption@Separator #2\par
1048   \else
1049     \global \@minipagefalse
1050     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
1051   \fi
1052   \vskip\belowcaptionskip}

```

Disable the standard warning with AMS and SMF classes.

```

1053 \@ifclassloaded{amsart}{\@FBwarning@capsepfalse}{}
1054 \@ifclassloaded{amsbook}{\@FBwarning@capsepfalse}{}
1055 \@ifclassloaded{amsdtx}{\@FBwarning@capsepfalse}{}
1056 \@ifclassloaded{amslatex}{\@FBwarning@capsepfalse}{}
1057 \@ifclassloaded{amproc}{\@FBwarning@capsepfalse}{}
1058 \@ifclassloaded{smfart}{\@FBwarning@capsepfalse}{}
1059 \@ifclassloaded{smfbook}{\@FBwarning@capsepfalse}{}

```

Disable the standard warning unless high punctuation is active.

```

1060 \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`).

```

1061 \newif\ifFB@koma
1062 \@ifclassloaded{memoir}{\@FBwarning@capsepfalse}{}
1063 \@ifclassloaded{scrartcl}{\@FBwarning@capsepfalse\FB@komatrue}{}
1064 \@ifclassloaded{scrbook}{\@FBwarning@capsepfalse\FB@komatrue}{}
1065 \@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).

```

1066 \@ifclassloaded{beamer}{\@FBwarning@capsepfalse}{}
1067 \ifdefined\@makecaption\else\@FBwarning@capsepfalse\fi

```

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

```

1068 \@ifpackageloaded{caption}
1069   {\FBWarning{frenchb.ldf}%
1070     {Please load the "caption" package\MessageBreak
1071       AFTER babel/frenchb; reported}%
1072   \@FBwarning@capsepfalse}%

```

```

1073     {}
    Same for package floatrow.
1074 \@ifpackageloaded{floatrow}
1075     {\FBWarning{frenchb.ldf}%
1076         {Please load the "floatrow" package\MessageBreak
1077             AFTER babel/frenchb; reported}%
1078     \@FBwarning@capsepfalse}%
1079     {}

```

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

```

1080 \AtBeginDocument{%
1081     \ifx\@makecaption\STD@makecaption
1082         \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 lan-
    guage.

1083     \ifFBoldFigTabCaptions
1084     \else
1085         \def\FBCaption@Separator{{\autospace@beforeFDP : }}%
1086     \fi
1087     \ifFBCustomiseFigTabCaptions
1088         \ifx\bbl@main@language\FB@french
1089             \def\FBCaption@Separator{\CaptionSeparator}%
1090         \fi
1091     \fi
1092     \@FBwarning@capsepfalse
1093 \fi
1094 \if@FBwarning@capsep
1095     \FBWarning{frenchb.ldf}%
1096     {Figures' and tables' captions might look like\MessageBreak
1097     'Figure 1:' which is wrong in French.\MessageBreak
1098     Check your class or packages to change this;\MessageBreak
1099     reported}%
1100 \fi
1101 \let\FB@makecaption\relax
1102 \let\STD@makecaption\relax
1103 }

```

2.8 Dots...

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

The `\if` construction in the $\text{\LaTeX}2_{\epsilon}$ definition of `\dots` doesn’t allow the use of `xspace` (`xspace` is always followed by a `\fi`), so we use the AMS- \LaTeX construction of `\dots`;

this has to be done ‘AtBeginDocument’ not to be overwritten when `amsmath.sty` is loaded after `babel`.

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

```
1104 \ifFBunicode
1105   \let\FBtextellipsis\textellipsis
1106 \else
1107   \DeclareTextSymbol{\FBtextellipsis}{LY1}{133}
1108   \DeclareTextCommandDefault{\FBtextellipsis}{%
1109     .\kern\fontdimen3\font.\kern\fontdimen3\font.\xspace}
1110 \fi
```

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

```
1111 \newcommand*{\Tdots@}{\@xp\textellipsis}
1112 \newcommand*{\Mdots@}{\@xp\mdots}
1113 \AtBeginDocument{\DeclareRobustCommand*{\dots}{\relax
1114                   \csname\ifmmode M\else T\fi dots@\endcsname}%
1115                   \ifdefined\@xp\else\let\@xp\relax\fi
1116                   \ifdefined\mdots\else\let\Mdots@\mathellipsis\fi
1117                   }
1118 \def\bbl@frenchdots{\babel@save\Tdots@ \let\Tdots@\FBtextellipsis}
1119 \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/frenchb` due to active characters issues (pdfLaTeX only).

```
1120 \ifFB@active@punct
1121   \@ifpackageloaded{listings}
1122     {\FBWarning{frenchb.ldf}%
1123       {Please load the "listings" package\MessageBreak
1124         AFTER babel/frenchb; reported}%
1125     }{}
1126 \fi
```

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

```
1127 \newif\if@FBwarning@natbib
1128 \ifFB@active@punct
1129   \@ifpackageloaded{natbib}{\@FBwarning@natbibtrue}
1130 \fi
1131 \AtBeginDocument{%
1132   \if@FBwarning@natbib
1133     \@ifpackageloaded{natbib}{\@FBwarning@natbibfalse}%
1134   \fi
```

```

1135 \if@FBwarning@natbib
1136 \FBWarning{frenchb.ldb}%
1137 {Please load the "natbib" package\MessageBreak
1138 BEFORE babel/frenchb; reported}%
1139 \fi
1140 }

```

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 frenchb ‘AtBeginDocument’. Reexecuting `\extrasfrench{}` is a possibility which was used up to v2.6h, it has been dropped in v3.0a because of its side-effects (f.i. `\babel@save` and `\babel@savevariable` no longer work).

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

```

1141 \newcommand*{\frenchbsetup}[1]{%
1142 \setkeys{FB}{#1}%
1143 }%
1144 \@onlypreamble\frenchbsetup

```

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

```

1145 \newif\ifFBShowOptions \FBShowOptionsfalse
1146 \newif\ifFBStandardLayout \FBStandardLayouttrue
1147 \newif\ifFBGlobalLayoutFrench \FBGlobalLayoutFrenchtrue
1148 \newif\ifFBReduceListSpacing \FBReduceListSpacingfalse
1149 \newif\ifFBListOldLayout \FBListOldLayoutfalse
1150 \newif\ifFBCompactItemize \FBCompactItemizefalse
1151 \newif\ifFBStandardItemizeEnv \FBStandardItemizeEnvtrue
1152 \newif\ifFBStandardEnumerateEnv \FBStandardEnumerateEnvtrue
1153 \newif\ifFBStandardItemLabels \FBStandardItemLabelstrue
1154 \newif\ifFBStandardLists \FBStandardListstrue
1155 \newif\ifFBIndentFirst \FBIndentFirstfalse
1156 \newif\ifFBFrenchFootnotes \FBFrenchFootnotesfalse
1157 \newif\ifFBAutoSpaceFootnotes \FBAutoSpaceFootnotesfalse
1158 \newif\ifFBOriginalTypewriter \FBOriginalTypewriterfalse
1159 \newif\ifFBThinColonSpace \FBThinColonSpacefalse
1160 \newif\ifFBThinSpaceInFrenchNumbers \FBThinSpaceInFrenchNumbersfalse
1161 \newif\ifFBFrenchSuperscripts \FBFrenchSuperscriptstrue
1162 \newif\ifFBLowercaseSuperscripts \FBLowercaseSuperscriptstrue

```

```

1163 \newif\ifFBPartNameFull           \FBPartNameFulltrue
1164 \newif\ifFBCustomiseFigTabCaptions \FBCustomiseFigTabCaptionsfalse
1165 \newif\ifFBOldFigTabCaptions       \FBOldFigTabCaptionsfalse
1166 \newif\ifFBSmallCapsFigTabCaptions \FBSmallCapsFigTabCaptionstrue
1167 \newif\ifFBSuppressWarning           \FBSuppressWarningfalse
1168 \newif\ifFBINGuillSpace              \FBINGuillSpacefalse

```

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

```

1169 \edef\FB@french{\CurrentOption}
1170 \AtEndOfPackage{%
1171   \ifx\bbl@main@language\FB@french
1172     \FBGlobalLayoutFrenchtrue
1173     \@ifclassloaded{beamer}%
1174       {\PackageInfo{frenchb.ldf}{%
1175         No list customisation for the beamer class,%
1176         \MessageBreak reported}}%
1177       {\FBReduceListSpacingtrue
1178        \FBStandardItemizeEnvfalse
1179        \FBStandardEnumerateEnvfalse
1180        \FBStandardItemLabelsfalse}%
1181     \FBIndentFirsttrue
1182     \FBFrenchFootnotesttrue
1183     \FBAutoSpaceFootnotesttrue
1184     \FBCustomiseFigTabCaptionstrue
1185   \else
1186     \FBGlobalLayoutFrenchfalse
1187   \fi

```

frenchb 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.

```

1188 \RequirePackage{keyval}%
1189 \define@key{FB}{ShowOptions}[true]%
1190   {\csname FBShowOptions#1\endcsname}%
1191 \define@key{FB}{StandardLayout}[true]%
1192   {\csname FBStandardLayout#1\endcsname
1193    \ifFBStandardLayout
1194      \FBReduceListSpacingfalse
1195      \FBStandardItemizeEnvtrue
1196      \FBStandardItemLabelstrue
1197      \FBStandardEnumerateEnvtrue
1198      \FBIndentFirstfalse
1199      \FBFrenchFootnotesfalse
1200      \FBAutoSpaceFootnotesfalse

```



```

1201             \FBGlobalLayoutFrenchfalse
1202         \else
1203             \FBReduceListSpacingtrue
1204             \FBStandardItemizeEnvfalse
1205             \FBStandardItemLabelsfalse
1206             \FBStandardEnumerateEnvfalse
1207             \FBIndentFirsttrue
1208             \FBFrenchFootnotesttrue
1209             \FBAutoSpaceFootnotesttrue
1210         \fi}%
1211 \define@key{FB}{GlobalLayoutFrench}[true]%
1212         {\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.

```

1213         \ifFBGlobalLayoutFrench
1214             \ifx\bbbl@main@language\FB@french
1215             \else
1216                 \PackageWarning{frenchb.ldf}%
1217                     {Option 'GlobalLayoutFrench' skipped:%
1218                     \MessageBreak French is *not*
1219                     babel's last option.\MessageBreak}%
1220             \fi
1221         \fi}%
1222 \define@key{FB}{ReduceListSpacing}[true]%
1223         {\csname FBReduceListSpacing#1\endcsname}%
1224 \define@key{FB}{ListOldLayout}[true]%
1225         {\csname FBListOldLayout#1\endcsname
1226         \ifFBListOldLayout
1227             \FBStandardEnumerateEnvtrue
1228             \renewcommand*{\FrenchLabelItem}{\textendash}%
1229         \fi}%
1230 \define@key{FB}{CompactItemize}[true]%
1231         {\csname FBCompactItemize#1\endcsname
1232         \ifFBCompactItemize
1233             \FBStandardItemizeEnvfalse
1234             \FBStandardEnumerateEnvfalse
1235         \else
1236             \FBStandardItemizeEnvtrue
1237             \FBStandardEnumerateEnvtrue
1238         \fi}%
1239 \define@key{FB}{StandardItemizeEnv}[true]%
1240         {\csname FBStandardItemizeEnv#1\endcsname}%
1241 \define@key{FB}{StandardEnumerateEnv}[true]%
1242         {\csname FBStandardEnumerateEnv#1\endcsname}%
1243 \define@key{FB}{StandardItemLabels}[true]%
1244         {\csname FBStandardItemLabels#1\endcsname}%
1245 \define@key{FB}{ItemLabels}{%
1246     \renewcommand*{\FrenchLabelItem}{#1}}%
1247 \define@key{FB}{ItemLabeli}{%

```

```

1248     \renewcommand*{\Frlabelitemi}{#1}%
1249 \define@key{FB}{ItemLabelii}{%
1250     \renewcommand*{\Frlabelitemii}{#1}%
1251 \define@key{FB}{ItemLabeliii}{%
1252     \renewcommand*{\Frlabelitemiii}{#1}%
1253 \define@key{FB}{ItemLabeliv}{%
1254     \renewcommand*{\Frlabelitemiv}{#1}%
1255 \define@key{FB}{StandardLists}[true]%
1256     {\csname FBStandardLists#1\endcsname
1257     \ifFBStandardLists
1258         \FBReduceListSpacingfalse
1259         \FBCompactItemizefalse
1260         \FBStandardItemizeEnvtrue
1261         \FBStandardEnumerateEnvtrue
1262         \FBStandardItemLabelstrue
1263     \else
1264         \FBReduceListSpacingtrue
1265         \FBCompactItemizetrue
1266         \FBStandardItemizeEnvfalse
1267         \FBStandardEnumerateEnvfalse
1268         \FBStandardItemLabelsfalse
1269     \fi}%
1270 \define@key{FB}{IndentFirst}[true]%
1271     {\csname FBIndentFirst#1\endcsname}%
1272 \define@key{FB}{FrenchFootnotes}[true]%
1273     {\csname FBFrenchFootnotes#1\endcsname}%
1274 \define@key{FB}{AutoSpaceFootnotes}[true]%
1275     {\csname FBAutoSpaceFootnotes#1\endcsname}%
1276 \define@key{FB}{AutoSpacePunctuation}[true]%
1277     {\csname FBAutoSpacePunctuation#1\endcsname}%
1278 \define@key{FB}{OriginalTypewriter}[true]%
1279     {\csname FBOriginalTypewriter#1\endcsname}%
1280 \define@key{FB}{ThinColonSpace}[true]%
1281     {\csname FBThinColonSpace#1\endcsname}%
1282 \define@key{FB}{ThinSpaceInFrenchNumbers}[true]%
1283     {\csname FBThinSpaceInFrenchNumbers#1\endcsname}%
1284 \define@key{FB}{FrenchSuperscripts}[true]%
1285     {\csname FBFrenchSuperscripts#1\endcsname}%
1286 \define@key{FB}{LowercaseSuperscripts}[true]%
1287     {\csname FBLowercaseSuperscripts#1\endcsname}%
1288 \define@key{FB}{PartNameFull}[true]%
1289     {\csname FBPartNameFull#1\endcsname}%
1290 \define@key{FB}{CustomiseFigTabCaptions}[true]%
1291     {\csname FBCustomiseFigTabCaptions#1\endcsname}%
1292 \define@key{FB}{OldFigTabCaptions}[true]%
1293     {\csname FBOldFigTabCaptions#1\endcsname}
    \CurrentOption no longer defined. It's value has been saved in \FB@CurOpt while
    reading frenchb.ldf.
1294     \ifFBOldFigTabCaptions
1295         \FB@addto{extras}{\babel@save\FBCaption@Separator

```

```

1296             \def\FBCaption@Separator{\CaptionSeparator}}%
1297         \fi}%
1298     \define@key{FB}{SmallCapsFigTabCaptions}[true]%
1299         {\csname FBSmallCapsFigTabCaptions#1\endcsname
1300         \ifFBSmallCapsFigTabCaptions
1301             \let\FBfigtabshape\scshape
1302         \else
1303             \let\FBfigtabshape\relax
1304         \fi}%
1305     \define@key{FB}{SuppressWarning}[true]%
1306         {\csname FBSuppressWarning#1\endcsname
1307         \ifFBSuppressWarning
1308             \renewcommand{\FBWarning}[2]{\relax}%
1309         \fi}%

```

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

```

1310     \define@key{FB}{INGuillSpace}[true]%
1311         {\csname FBINGuillSpace#1\endcsname}%
1312     \define@key{FB}{InnerGuillSingle}[true]%
1313         {\csname FBInnerGuillSingle#1\endcsname}%
1314     \define@key{FB}{EveryParGuill}{\expandafter\let\expandafter
1315         \FBeveryparguill\csname FBguill#1\endcsname}%
1316     \define@key{FB}{EveryLineGuill}{\expandafter\let\expandafter
1317         \FBeverylineguill\csname FBguill#1\endcsname
1318         \ifFB@luatex@punct
1319         \else
1320             \let\FBeverylineguill\FBguillnone
1321             \PackageWarning{frenchb.ldb}%
1322                 {Option ‘EveryLineGuill’ skipped:%
1323                 \MessageBreak this option is for
1324                 LuaTeX *only*.\MessageBreak Reported}%
1325         \fi}%

```

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

```

1326     \define@key{FB}{og}{%
1327         \ifFBunicode

```

LuaTeX or XeTeX in use, first try modern LuaTeX: we just need to set LuaTeX's attribute

```

\FB@addGUILspace to 1,
1328         \ifFB@luatex@punct
1329         \FB@addGUILspace=1 \relax
1330         \fi

then with XeTeX it is a bit more tricky:
1331         \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 subsec-
tion 2.2).
1332         \XeTeXcharclass"13 = \FB@guilo
1333         \XeTeXcharclass"AB = \FB@guilo
1334         \XeTeXcharclass"A0 = \FB@guilnul
1335         \XeTeXcharclass"202F = \FB@guilnul
1336         \fi
1337         \else

This is for conventional TeX engines:
1338         \newcommand*{\FB@@og}{%
1339         \iflanguage{french}%
1340         {\ifFBAutoSpaceGuill\FB@og\ignorespaces
1341         \else\guillemotleft
1342         \fi}%
1343         {\guillemotleft}}%
1344         \AtBeginDocument{%
1345         \ifdefined\DeclareInputText
1346         \ifdefined\uc@dc lc

Package inputenc with utf8x encoding loaded, use \uc@dc lc,
1347         \uc@dc lc{171}{default}{\FB@@og}%
1348         \else

if encoding is not utf8x, try utf8...
1349         \ifdefined\DeclareUnicodeCharacter

utf8 loaded, use \DeclareUnicodeCharacter,
1350         \DeclareUnicodeCharacter{00AB}{\FB@@og}%
1351         \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.
1352         \@tempcnta'#1\relax
1353         \ifdefined\mule@def
1354         \mule@def{11}{\FB@@og}%
1355         \else
1356         \DeclareInputText{\the\@tempcnta}{\FB@@og}%
1357         \fi
1358         \fi
1359         \fi
1360         \else

Package inputenc not loaded, no way...
1361         \PackageWarning{frenchb.ldf}%

```

```

1362             {Option 'og' requires package inputenc.\MessageBreak}%
1363         \fi
1364     }%
1365 \fi
1366 }%

    Same code for the closing quote.
1367 \define@key{FB}{fg}{%
1368     \ifFBunicode
1369         \ifFB@luatex@punct
1370             \FB@addGUILspace=1 \relax
1371         \fi
1372         \ifFB@xetex@punct
1373             \XeTeXcharclass"14 = \FB@guilf
1374             \XeTeXcharclass"BB = \FB@guilf
1375             \XeTeXcharclass"A0 = \FB@guilnul
1376             \XeTeXcharclass"202F = \FB@guilnul
1377         \fi
1378     \else
1379         \newcommand*{\FB@@fg}{%
1380             \iflanguage{french}%
1381                 {\ifFBAutoSpaceGuill\FB@fg
1382                 \else\guillemotright
1383                 \fi}%
1384                 {\guillemotright}}%
1385         \AtBeginDocument{%
1386             \ifdefined\DeclareInputText
1387                 \ifdefined\uc@dcl
1388                     \uc@dcl{187}{default}{\FB@@fg}%
1389                 \else
1390                     \ifdefined\DeclareUnicodeCharacter
1391                         \DeclareUnicodeCharacter{00BB}{\FB@@fg}%
1392                     \else
1393                         \@tempcnta'1\relax
1394                         \ifdefined\mule@def
1395                             \mule@def{27}{\FB@@fg}%
1396                         \else
1397                             \DeclareInputText{\the\@tempcnta}{\FB@@fg}%
1398                         \fi
1399                     \fi
1400                 \fi
1401             \else
1402                 \PackageWarning{frenchb.ldf}%
1403                     {Option 'fg' requires package inputenc.\MessageBreak}%
1404             \fi
1405         }%
1406     \fi
1407 }%
1408 }

```

\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: cur-

rently 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`.

```
1409 \newcommand*{\FBprocess@options}{%
```

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

```
1410 \ifpackageloaded{enumitem}{%
1411   \ifFBStandardItemizeEnv
1412   \else
1413     \FBStandardItemizeEnvtrue
1414     \PackageInfo{frenchb.ldb}{%
1415       {Setting StandardItemizeEnv=true for\MessageBreak
1416         compatibility with enumitem package,\MessageBreak}%
1417     \fi
1418   \ifFBStandardEnumerateEnv
1419   \else
1420     \FBStandardEnumerateEnvtrue
1421     \PackageInfo{frenchb.ldb}{%
1422       {Setting StandardEnumerateEnv=true for\MessageBreak
1423         compatibility with enumitem package,\MessageBreak}%
1424     \fi}}%
1425 \ifpackageloaded{paralist}{%
1426   \ifFBStandardItemizeEnv
1427   \else
1428     \FBStandardItemizeEnvtrue
1429     \PackageInfo{frenchb.ldb}{%
1430       {Setting StandardItemizeEnv=true for\MessageBreak
1431         compatibility with paralist package,\MessageBreak}%
1432     \fi
1433   \ifFBStandardEnumerateEnv
1434   \else
1435     \FBStandardEnumerateEnvtrue
1436     \PackageInfo{frenchb.ldb}{%
1437       {Setting StandardEnumerateEnv=true for\MessageBreak
1438         compatibility with paralist package,\MessageBreak}%
1439     \fi}}%
1440 \ifpackageloaded{enumerate}{%
1441   \ifFBStandardEnumerateEnv
1442   \else
1443     \FBStandardEnumerateEnvtrue
1444     \PackageInfo{frenchb.ldb}{%
1445       {Setting StandardEnumerateEnv=true for\MessageBreak
1446         compatibility with enumerate package,\MessageBreak}%
1447     \fi}}%
```

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

```
1448 \def\FB@ufl{\update@frenchlists}
1449 \ifx\bbl@main@language\FB@french
```

```

1450 \update@frenchlists
1451 \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.

```

1452 \ifFBAutoSpacePunctuation
1453 \autospace@beforeFDP
1454 \else
1455 \noautospace@beforeFDP
1456 \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.

```

1457 \ifFBOriginalTypewriter
1458 \else
1459 \let\ttfamilyORI\ttfamily
1460 \let\rmfamilyORI\rmfamily
1461 \let\sffamilyORI\sffamily
1462 \let\ttfamily\ttfamilyFB
1463 \let\rmfamily\rmfamilyFB
1464 \let\sffamily\sffamilyFB
1465 \fi

```

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

```

1466 \ifFBThinColonSpace
1467 \ifFB@luatex@punct
1468 \FBcolonskip=\FBthinskip\relax
1469 \else
1470 \renewcommand*{\FBcolonspace}{\FBthinspace}%
1471 \fi
1472 \fi

```

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

```

1473 \ifFBINGuillSpace
1474 \ifFB@luatex@punct
1475 \FBguillskip=3.33pt plus 1.665pt minus 1.11pt \relax
1476 \else
1477 \renewcommand*{\FBguillspace}{\space}%
1478 \fi
1479 \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.

```

1480 \@ifpackageloaded{numprint}%
1481 {\ifnprt@autolanguage
1482   \providecommand*\npstylefrench{}}%
1483   \ifFBThinSpaceInFrenchNumbers
1484     \renewcommand*\npstylefrench{%
1485       \npthousandsep{\,}%
1486       \npdecimalsign{,}%
1487       \npproductsign{\cdot}%
1488       \npunitseparator{\,}%
1489       \npdegreeseperator{}}%
1490       \nppercentseparator{\nprt@unitsep}%
1491     }%
1492   \else
1493     \renewcommand*\npstylefrench{%
1494       \npthousandsep{~}%
1495       \npdecimalsign{,}%
1496       \npproductsign{\cdot}%
1497       \npunitseparator{\,}%
1498       \npdegreeseperator{}}%
1499       \nppercentseparator{\nprt@unitsep}%
1500     }%
1501   \fi
1502   \npaddtolanguage{french}{french}%
1503 \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.

```

1504 \ifFBFrenchSuperscripts
1505   \DeclareRobustCommand*\up*{\@ifstar{\FB@up@fake}{\fup}}%
1506 \else
1507   \DeclareRobustCommand*\up*{\@ifstar{\FB@up@fake}%
1508                                   {\textsuperscript}}%
1509 \fi

```

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

```

1510 \ifFBLowercaseSuperscripts
1511 \else
1512   \renewcommand*\FB@lc[1]{##1}%
1513 \fi

```

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

```

1514 \ifFBCustomiseFigTabCaptions
1515   \ifFB@koma
1516     \renewcommand*\captionformat{\CaptionSeparator}%
1517   \fi
1518   \@ifclassloaded{memoir}%
1519     {\captiondelim{\CaptionSeparator}}{}%
1520   \@ifclassloaded{beamer}%
1521     {\defbeamertemplate{caption label separator}{FBcustom}}{}%

```



```

1522         \CaptionSeparator}%
1523         \setbeamertemplate{caption label separator}{FBcustom}}{}%
1524     \else

```

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

```

1525     \ifFB@koma
1526         \renewcommand*{\captionformat}{\autospace@beforeFDP : }%
1527     \fi
1528     \@ifclassloaded{memoir}%
1529         {\captiondelim{\autospace@beforeFDP : }}%
1530     }{}%
1531     \@ifclassloaded{beamer}%
1532         {\defbeamertemplate{caption label separator}{FBcolon}{%
1533             {\autospace@beforeFDP : }}%
1534             \setbeamertemplate{caption label separator}{FBcolon}%
1535         }{}%
1536     \fi

```

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

```

1537 \ifFBShowOptions
1538     \GenericWarning{* }{%
1539         * **** List of possible options for frenchb ****\MessageBreak
1540         [Default values between brackets when frenchb is loaded *LAST*]%
1541         \MessageBreak
1542         ShowOptions=true [false]\MessageBreak
1543         StandardLayout=true [false]\MessageBreak
1544         GlobalLayoutFrench=false [true]\MessageBreak
1545         StandardLists=true [false]\MessageBreak
1546         IndentFirst=false [true]\MessageBreak
1547         ReduceListSpacing=false [true]\MessageBreak
1548         ListOldLayout=true [false]\MessageBreak
1549         StandardItemizeEnv=true [false]\MessageBreak
1550         StandardEnumerateEnv=true [false]\MessageBreak
1551         StandardItemLabels=true [false]\MessageBreak
1552         ItemLabels=\textendash, \textbullet,
1553             \protect\ding{43},... [\textendash]\MessageBreak
1554         ItemLabeli=\textendash, \textbullet,
1555             \protect\ding{43},... [\textendash]\MessageBreak
1556         ItemLabelii=\textendash, \textbullet,
1557             \protect\ding{43},... [\textendash]\MessageBreak
1558         ItemLabeliii=\textendash, \textbullet,
1559             \protect\ding{43},... [\textendash]\MessageBreak
1560         ItemLabeliv=\textendash, \textbullet,
1561             \protect\ding{43},... [\textendash]\MessageBreak
1562         FrenchFootnotes=false [true]\MessageBreak
1563         AutoSpaceFootnotes=false [true]\MessageBreak
1564         AutoSpacePunctuation=false [true]\MessageBreak
1565         OriginalTypewriter=true [false]\MessageBreak
1566         ThinColonSpace=true [false]\MessageBreak
1567         ThinSpaceInFrenchNumbers=true [false]\MessageBreak
1568         FrenchSuperscripts=false [true]\MessageBreak

```

```

1569 LowercaseSuperscripts=false [true]\MessageBreak
1570 PartNameFull=false [true]\MessageBreak
1571 SuppressWarning=true [false]\MessageBreak
1572 CustomiseFigTabCaptions=false [true]\MessageBreak
1573 OldFigTabCaptions=true [false]\MessageBreak
1574 SmallCapsFigTabCaptions=false [true]\MessageBreak
1575 INGuillSpace=true [false]\MessageBreak
1576 InnerGuillSingle=true [false]\MessageBreak
1577 EveryParGuill=open, close, none [open]\MessageBreak
1578 EveryLineGuill=open, close, none
1579         [open in LuaTeX, none otherwise]\MessageBreak
1580 og= <left quote character>, fg= <right quote character>%
1581 \MessageBreak
1582 *****%
1583 \MessageBreak\protect\frenchbsetup{ShowOptions}}
1584 \fi
1585 }

```

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.

```

1586 \AtBeginDocument{%
1587   \providecommand*{\xspace}{\relax}%

```

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

```

1588   \ifdefined\pdfstringdefDisableCommands
1589     \pdfstringdefDisableCommands{%
1590       \let\up\relax
1591       \let\fu\relax
1592       \let\degre\textdegree
1593       \let\degres\textdegree
1594       \def\ieme{e\xspace}%
1595       \def\iemes{es\xspace}%
1596       \def\ier{er\xspace}%
1597       \def\iers{ers\xspace}%
1598       \def\iere{re\xspace}%
1599       \def\ieres{res\xspace}%
1600       \def\FrenchEnumerate#1{#1\degre\space}%
1601       \def\FrenchPopularEnumerate#1{#1\degre)\space}%
1602       \def\No{N\degre\space}%
1603       \def\no{n\degre\space}%
1604       \def\Nos{N\degre\space}%
1605       \def\nos{n\degre\space}%
1606       \def\FB@og{\guillemotleft\space}%
1607       \def\FB@fg{\space\guillemotright}%
1608       \def\at{@}%
1609       \def\circonflexe{\string^}%
1610       \def\tild{\string~}%
1611       \let\bsc\textsc
1612     }%

```

```

1613   \fi

    It is time to process the options set with \frenchbsetup{} or later.
1614   \FBprocess@options

    With LuaTeX engines (\FBthinskip and \FBcolonskip values are set now), it is time
    to load file frenchb.lua.
1615   \ifFB@luatex@punct
1616       \activate@luatexpunct
1617   \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!
1618   \ifFBunicode
1619       \ifdefined\DeclareUTFcharacter
1620       \else
1621           \@ifpackageloaded{luainputenc}{}%
1622           {\PackageWarning{frenchb.ldb}%
1623             {Add \protect\usepackage{fontspec} to the\MessageBreak
1624              preamble of your document,}%
1625           }%
1626       \fi
1627   \else
1628       \begingroup \newcommand{\FBOTone}{OT1}%
1629       \ifx\encodingdefault\FBOTone
1630           \PackageWarning{frenchb.ldb}%
1631             {OT1 encoding should not be used for French.%
1632              \MessageBreak
1633              Add \protect\usepackage[T1]{fontenc} to the
1634              preamble\MessageBreak of your document,}%
1635       \fi
1636       \endgroup
1637   \fi
1638 }

```

2.11 French lists

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

The amount of vertical space before and after a list is given by `\topsep` + `\parskip` (+ `\partopsep` if the list starts a new paragraph). IMHO, `\parskip` should be added *only* when the list starts a new paragraph, so I subtract `\parskip` from `\topsep` and

add it back to `\partopsep`; this will normally make no difference because `\parskip`'s default value is 0pt, but will be noticeable when `\parskip` is *not* null.

```

1639 \let\listORI\list
1640 \let\endlistORI\endlist
1641 \def\FB@listVsettings{%
1642     \setlength{\itemsep}{0.4ex plus 0.2ex minus 0.2ex}%
1643     \setlength{\parsep}{0.4ex plus 0.2ex minus 0.2ex}%
1644     \setlength{\topsep}{0.8ex plus 0.4ex minus 0.4ex}%
1645     \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.
1646     \@tempdima=\parskip
1647     \addtolength{\topsep}{-\@tempdima}%
1648     \addtolength{\partopsep}{\@tempdima}%
1649 }
1650 \def\listFB#1#2{\listORI{#1}{\FB@listVsettings #2}}
1651 \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:

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

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

```

\FrenchLabelItem \newcommand*{\FrenchLabelItem}{\textemdash}
\FrenchLabelItemii \newcommand*{\FrenchLabelItemii}{\FrenchLabelItem}
\FrenchLabelItemiii \newcommand*{\FrenchLabelItemiii}{\FrenchLabelItem}
\FrenchLabelItemiv \newcommand*{\FrenchLabelItemiv}{\FrenchLabelItem}
\FrenchLabelItemv \newcommand*{\FrenchLabelItemv}{\FrenchLabelItem}

```

`\listindentFB` Let's define two lengths `\listindentFB` and `\labelwidthFB` to customise lists' horizontal indentations. They are given silly values here (–1 pt) in order to eventually enable their customisation in the preamble. They will get reasonable defaults

later when entering French (see `\bbl@frenchlabelitems`) unless they have been customised.

```
1657 \newlength\listindentFB
1658 \setlength{\listindentFB}{-1pt}
1659 \newlength\labelwidthFB
1660 \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.

```
1661 \newlength\leftmarginFB
1662 \def\FB@listHsettings{%
1663   \leftmarginFB\labelwidthFB
1664   \advance\leftmarginFB \labelsep
1665   \leftmargini\leftmarginFB
1666   \advance\leftmargini \listindentFB
1667   \leftmarginii\leftmarginFB
1668   \leftmarginiii\leftmarginFB
1669   \leftmarginiv\leftmarginFB
1670   \leftmargin\csname leftmargin\romannumeral\the\@listdepth\endcsname
1671 }
```

`\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).

```
1672 \def\FB@itemizesettings{%
1673   \ifFBReduceListSpacing
1674     \setlength{\itemsep}{\z@}%
1675     \setlength{\parsep}{\z@}%
1676     \setlength{\topsep}{\z@}%
1677     \setlength{\partopsep}{\z@}%
1678     \@tempdima=\parskip
1679     \addtolength{\topsep}{-\@tempdima}%
1680     \addtolength{\partopsep}{-\@tempdima}%
1681   \fi
1682   \settowidth{\labelwidth}{\csname\@itemitem\endcsname}%
1683   \ifBListOldLayout
1684     \setlength{\leftmargin}{\labelwidth}%
1685     \addtolength{\leftmargin}{\labelsep}%
1686     \addtolength{\leftmargin}{\parindent}%
1687   \else
1688     \FB@listHsettings
1689   \fi
1690 }
```

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`.

```
1691 \def\itemizeFB{%
```

```

1692 \ifnum \@itemdepth >\thr@@\toodeep\else
1693 \advance\@itemdepth\@ne
1694 \edef\@itemitem{labelitem\romannumeral\the\@itemdepth}%
1695 \expandafter
1696 \listORI
1697 \csname\@itemitem\endcsname
1698 \FB@itemizesettings
1699 \fi
1700 }
1701 \let\enditemizeFB\endlistORI

1702 \def\labelitemsFB{%
1703 \let\labelitemi\Frlabelitemi
1704 \let\labelitemii\Frlabelitemii
1705 \let\labelitemiii\Frlabelitemiii
1706 \let\labelitemiv\Frlabelitemiv
1707 \ifdim\labelwidthFB<\z@
1708 \settowidth{\labelwidthFB}{\FrenchLabelItem}%
1709 \fi
1710 \ifdim\listindentFB<\z@
1711 \ifdim\parindent=\z@
1712 \setlength{\listindentFB}{1.5em}%
1713 \else
1714 \setlength{\listindentFB}{\parindent}%
1715 \fi
1716 \fi
1717 }

```

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

```

1718 \def\enumerateFB{%
1719 \ifnum \@enumdepth >\thr@@\toodeep\else
1720 \advance\@enumdepth\@ne
1721 \edef\@enumctr{enum\romannumeral\the\@enumdepth}%
1722 \expandafter
1723 \list
1724 \csname label\@enumctr\endcsname
1725 {\FB@listHsettings
1726 \usecounter\@enumctr\def\makelabel##1{\hss\llap{##1}}}%
1727 \fi
1728 }
1729 \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).

```

1730 \def\descriptionFB{%
1731 \list{}{\FB@listHsettings

```

```

1732          \labelwidth\z@
1733          \itemindent-\leftmargin
1734          \ifnum\@listdepth=1
1735              \advance\itemindent by \listindentFB
1736          \fi
1737          \let\makelabel\descriptionlabel}%
1738 }
1739 \let\enddescriptionFB\endlistORI

```

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

```

\bbbl@frenchlistlayout 1740 \def\update@frenchlists{%
\bbbl@nonfrenchlistlayout 1741 \ifFBReduceListSpacing \let\list\listFB \fi
1742 \ifFBStandardItemizeEnv
1743 \else \let\itemize\itemizeFB \fi
1744 \ifFBStandardItemLabels
1745 \else \labelitemsFB \fi
1746 \ifFBStandardEnumerateEnv
1747 \else \let\enumerate\enumerateFB \let\description\descriptionFB \fi
1748 }

```

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. 54.

```

1749 \def\FB@ufl{\relax}
1750 \def\bbbl@frenchlistlayout{%
1751 \ifFBGlobalLayoutFrench
1752 \else
1753 \babel@save\list          \babel@save\itemize
1754 \babel@save\enumerate    \babel@save\description
1755 \babel@save\labelitemi   \babel@save\labelitemii
1756 \babel@save\labelitemiii \babel@save\labelitemiv
1757 \fi
1758 \FB@ufl
1759 }
1760 \def\bbbl@nonfrenchlistlayout{%
1761 \ifFBGlobalLayoutFrench
1762 \update@frenchlists
1763 \fi
1764 }
1765 \FB@addto{extras}{\bbbl@frenchlistlayout}
1766 \FB@addto{noextras}{\bbbl@nonfrenchlistlayout}

```

2.12 French indentation of sections

`\bbbl@frenchindent` In French the first paragraph of each section should be indented, this is another difference with US-English. This is controlled by the flag `\if@afterindent`.

We will need to save the value of the flag `\if@afterindent` ‘AtBeginDocument’ before eventually changing its value.

```

1767 \def\bbbl@frenchindent{%

```

```

1768 \ifFBGlobalLayoutFrench\else\babel@save\@afterindentfalse\fi
1769 \ifFBIndentFirst
1770     \let\@afterindentfalse\@afterindenttrue
1771     \@afterindenttrue
1772 \fi}
1773 \def\bbl@nonfrenchindent{%
1774     \ifFBGlobalLayoutFrench
1775         \ifFBIndentFirst
1776             \@afterindenttrue
1777         \fi
1778     \fi}
1779 \FB@addto{extras}{\bbl@frenchindent}
1780 \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 `frenchb` will drop the customisation of footnotes. The layout of footnotes is controlled by two flags `\ifFBAutoSpaceFootnotes` and `\ifFBFrenchFootnotes` which are set by options of `\frenchbsetup{}` (see section 2.10). 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!).

When `\ifFBAutoSpaceFootnotes` is true, `\@footnotemark` (the definition of which is saved at the `\begin{document}` in order to include any customisation that packages might have done) is redefined to add a thin space before the number or symbol calling a footnote (any space typed in is removed first). This has no effect on the layout of the footnote itself.

```

1781 \AtBeginDocument{\@ifpackageloaded{bigfoot}%
1782     {\PackageInfo{frenchb.ldb}%
1783     {bigfoot package in use.\MessageBreak
1784     frenchb will NOT customise footnotes;\MessageBreak
1785     reported}}%
1786     {\let\@footnotemarkORI\@footnotemark
1787     \def\@footnotemarkFB{\leavevmode\unskip\unkern
1788     \,\@footnotemarkORI}%
1789     \ifFBAutoSpaceFootnotes
1790         \let\@footnotemark\@footnotemarkFB
1791     \fi}%
1792 }

```

We then define `\@makefntextFB`, a variant of `\@makefntext` which is responsible for the layout of footnotes, to match the specifications of the French ‘Imprimerie Nationale’: footnotes will be indented by `\parindentFFN`, numbers (if any) typeset on the baseline (instead of superscripts) and followed by a dot and an half quad space. Whenever symbols are used to number footnotes (as in `\thanks` for instance), we switch back to the standard layout (the French layout of footnotes is meant for footnotes numbered by Arabic or Roman digits).

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

```

1793 \newcommand*{\dotFFN}{.}
1794 \newcommand*{\kernFFN}{\kern .5em}
1795 \newdimen\parindentFFN
1796 \parindentFFN=10in
1797 \def\ftnISSymbol{\@fnsymbol\c@footnote}
1798 \long\def\@makefntextFB#1{\ifx\thefootnote\ftnISSymbol
1799     \@makefntextORI{#1}%
1800     \else
1801     \parindent=\parindentFFN
1802     \rule{z}{\footnotesep}
1803     \setbox\@tempboxa\hbox{\@thefnmark}%
1804     \ifdim\wd\@tempboxa>z@
1805     \llap{\@thefnmark}\dotFFN\kernFFN
1806     \fi #1
1807     \fi}%

```

We save the standard definition of `\@makefntext` at the `\begin{document}`, and then redefine `\@makefntext` according to the value of flag `\ifFBFrenchFootnotes` (true or false).

```

1808 \AtBeginDocument{\@ifpackageloaded{bigfoot}{}%
1809     {\ifdim\parindentFFN<10in
1810     \else
1811     \parindentFFN=\parindent
1812     \ifdim\parindentFFN<1.5em \parindentFFN=1.5em \fi
1813     \fi
1814     \let\@makefntextORI\@makefntext
1815     \long\def\@makefntext#1{%
1816     \ifFBFrenchFootnotes
1817     \@makefntextFB{#1}%
1818     \else
1819     \@makefntextORI{#1}%
1820     \fi}%
1821     }%
1822 }

```

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

```

1823 \newcommand*{\AddThinSpaceBeforeFootnotes}{\FBAutoSpaceFootnotestruer}
1824 \newcommand*{\FrenchFootnotes}{\FBFrenchFootnotestruer}
1825 \newcommand*{\StandardFootnotes}{\FBFrenchFootnotesfalse}

```

2.14 Clean up and exit

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

```
1826 \FBclean@on@exit
1827 \let\FB@llc\loadlocalcfg
1828 \let\loadlocalcfg\@gobble
1829 \ldf@finish\CurrentOption
1830 \let\loadlocalcfg\FB@llc
```

3 Change History

v2.0	General: <code>\parindentFFN</code> not changed if already defined (required by JA for <code>cah-gut.cls</code>). 64	v2.0g	<code>\frenchbsetup</code> : Revert previous change to <code>StandardLayout</code> . This option must set the three flags <code>\FBReduceListSpacingfalse</code> , <code>\FBCompactItemizefalse</code> , and <code>\FBStandardItemLabeltrue</code> instead of <code>\FBStandardListstrue</code> , so that later options can still change their value before executing <code>\FBprocess@options</code> . Same thing for option <code>StandardLists</code> . . . 47
	Added warning for OT1 encoding. 58		
	Footnotes are now printed by default ‘à la française’ for the whole document. 64	v2.1a	General: Command <code>\fup</code> added to produce better superscripts than <code>\textsuperscript</code> 35
	New command <code>\frenchbsetup</code> added for global customisation. . 47		<code>\datefrench</code> : <code>\today</code> changed (correction in 2.0 was wrong: <code>\today</code> was printed without spaces in toc). 35
	<code>\bsc</code> : <code>\hbox</code> dropped, replaced by <code>\kern0pt</code> 38		<code>\frenchbsetup</code> : New option: French-Superscripts to define <code>\up</code> as <code>\fup</code> or as <code>\textsuperscript</code> 47
	<code>\captionsfrench</code> : ‘Fig.’ changed to ‘Figure’ and ‘Tab.’ to ‘Table’. . . . 41		New option: LowercaseSuperscripts. 47
	<code>\datefrench</code> : 2 ‘ <code>\relax</code> ’ added in <code>\today</code> ’s definition. 35	v2.1b	General: Disable some commands in bookmarks. 58
	<code>\FBtextellipsis</code> : Added special case for LY1 encoding, see bug report from Bruno Voisin (2004/05/18). 46		<code>\fup</code> : Command <code>\fup</code> changed to use real superscripts from fourier v. 1.6. 35
	<code>\nombre</code> : <code>\nombre</code> now requires <code>numprint.sty</code> 40	v2.1c	General: Added commands <code>\Nos</code> and <code>\nos</code> 37
v2.0b	General: Footnotes: Just do nothing (except warning) when the bigfoot package is loaded. 64		<code>\degres</code> : Provide a temporary definition (hyperref safe) of <code>\degres</code> in case it has to be expanded in the preamble (by beamer’s <code>\title</code> command for instance). 39
v2.0c	General: There is no need to define here <code>numprint</code> ’s command <code>\npstylefrench</code> , it will be re-defined ‘ <code>AtBeginDocument</code> ’ by <code>\FBprocess@options</code> 41		<code>\up</code> : Provide a temporary definition (hyperref safe) of <code>\up</code> in case it has to be expanded in the preamble (by beamer’s <code>\title</code> command for instance). 35
	<code>\frenchbsetup</code> : Option <code>ThinSpaceInFrenchNumbers</code> added. 47	v2.1d	General: Argument of <code>\ProvidesLanguage</code> changed above from ‘french’ to ‘frenchb’ (otherwise <code>\listfiles</code> prints no date/version information). The real name of current language
v2.0d	<code>\frenchbsetup</code> : Options <code>og</code> and <code>fg</code> changed: limit the definition to French so that quote characters can be used in German. 47		
v2.0e	<code>\frenchbsetup</code> : New option: <code>StandardLists</code> 47		
v2.0f	<code>\frenchbsetup</code> : <code>StandardLayout</code> option had no effect on lists. Test moved to <code>\FBprocess@options</code> . 47		
	Two typos corrected in option <code>StandardLists</code> : <code>[false]</code> → <code>[true]</code> and <code>StandardLayout</code> → <code>StandardLists</code> . 47		

(french) as to be corrected before calling \LdfInit.	13	v2.3c	\ttfamilyFB: Commands \ttfamily, \rmfamily and \sffamily have to be robust. Bug introduced in 2.3a, pointed out by Manuel Pégourié-Gonnard.	30
Avoid warning “\end occurred when \ifx ... incomplete” with LaTeX-2.09.	13			
v2.2a		v2.3d	\bbl@nonfrenchindent: Bug correction: previous versions of frenchb set the flag \if@afterindent to false outside French which is correct for English but wrong for some languages like Spanish. Pointed out by Juan José Torrens.	63
\frenchbsetup: Default values of flags changed: default now means ‘StandardLayout’, they will be changed to ‘FrenchLayout’ AtEndOfPackage only if french is \bbl@main@language.	47	v2.3e	\NoAutoSpaceBeforeFDP: Execute \AutoSpaceBeforeFDP also in LaTeX to define \FDP@colonspace: needed for tex4ht, pointed out by MPG.	29
The global layout of the document is no longer changed when frenchb is not the last option of babel (\bbl@main@language). Suggested by Ulrike Fischer.	47	v2.4a	General: \PackageWarning changed to \FBWarning (when bigfoot package in use).	64
When frenchb is babel’s last option, French becomes the document’s main language, so GlobalLayout-French applies.	47	\CaptionSeparator: \PackageWarning changed to \FBWarning (in case \@makecaption has been customised). \FBWarning is defined as \PackageWarning by default but can be made silent using \frenchbsetup, (suggested by MPG).	43	
\fup: \newif and \newdimen moved before \ifLaTeXe to avoid an error with plainTeX.	35	\frenchbsetup: New option SuppressWarning.	47	
v2.3a		\ifFBXeTeX: Added a new ‘if’ \FBunicode and some \lccode definitions to \extrasfrench and \noextrasfrench.	15	
General: In LaTeX, frenchb no longer adds spaces before ‘high punctuation’ characters in computer code. Suggested by Yannis Haralambous.	30	v2.4c	General: In \ttfamilyFB, also cancel automatic spaces inside French guillemets coded as characters (see \frenchbsetup).	51
\frenchbsetup: New option: OriginalTypewriter. Now frenchb switches to \noautospace@beforeFDP when a tt-font is in use. When OriginalTypewriter is set to true, frenchb behaves as in pre-2.3 versions.	47	\ttfamilyFB: In \ttfamilyFB, also cancel automatic spaces inside French guillemets entered as characters (see \frenchbsetup). ...	30	
\fup: \lowercase changed to \MakeLowercase as the former doesn’t work for non ASCII characters in encodings like applemac, utf-8,...	35	v2.4d	\up: Command \up defined with \providecommand instead of \newcommand as \up may be de-	
\NoAutoSpaceBeforeFDP: \NoAutoSpaceBeforeFDP and \AutoSpaceBeforeFDP now set the flag \ifFBAutoSpacePunctuation accordingly (LaTeX only).	29			
v2.3b				
General: New commands \dotFFN and \kernFFN for more flexibility (suggested by JA).	64			

finned elsewhere (catalan.ldf). Bug pointed out by Felip Manyé i Ballester.	35	\FBXeTeX and \FBLuaTeX as XeTeX and behave differently regarding the status of the French “apostrophe”.	15
v2.5a		v2.5e	
General: \og and \fg do not print correctly in English when using XeTeX or LuaTeX, fixed by using \textquotedblleft and \textquotedblright defined above.	32	General: \pdfstringdefDisableCommands should redefine \FB@og and \FB@fg instead of \og and \fg so that it works also when quotes are entered as characters. Reported by Sébastien Gouezel.	58
New command \NoAutoSpacing, suggested by MPG.	30	v2.5f	
Punctuation is no longer made active with XeTeX-based engines.	16	General: Changed definitions of \at, \circflex, \tild, \boi and \deg for Unicode based engines.	38
\captionsfrench: \emph deleted in \seename and \alsoname to match what is done for the other languages. Suggested by Marc Baudoin.	41	\FBtextellipsis: Unicode fonts also provide a ready made character for \textellipsis, let’s just use it (reported by Maxime Chupin, 2011/06/04).	46
\FBthinspace: Define \FBthinspace for those who want to customise the width of the space before ; and co.	17	v2.5g	
\textquotedblright: Change \guillemotleft and \guillemotright definitions for Unicode and provide definitions for \textquotedblleft and \textquotedblright. Insures correct printing of quotes by \og and \fg in French and outside.	31	General: Redefine \deg, \degres \at \circflex and \tild for bookmarks. Add \fup also.	58
v2.5b		When \ifFB@xetex@punct is true, ‘og’ and ‘fg’ options now set XeTeX-charclasses of these characters to \FB@guilo and \FB@guilf. Otherwise French quotes behave as normal characters (their XeTeXcharclass is 0).	51
General: Do not use the test \iflanguage{french} to check whether French is the main language or not, as it might be erroneously positive when English is the main language and no hyphenation patterns are available for French. In this case \l@french and \l@english are 0. Pointed out by Günter Milde.	48	\FB@xetex@punct@french: XeTeX-charclass(es) for French quotes will be set to \FB@guilo and \FB@guilf by options ‘og’ and ‘fg’ in \frenchbsetup. French quotes should behave as normal characters by default in XeLaTeX as in LaTeX.	25
v2.5c		v2.5h	
General: The code meant for XeTeX also works for LuaTeX, we just need to change the test.	51	\degres: textcomp.sty has changed. The test about \M@TS1 is no longer relevant, let’s change it.	39
v2.5d		v2.5i	
General: Moved the \newcount command outside \ifFB@xetex@punct ... \fi (it broke Plain formats). ..	25	General: Temporary fix: as long as xeCJK.sty will not use \newXeTeXintercharclass to allocate its classes, we will have to define 3 fake classes.	25
\ifFBXeTeX: Added two new ‘if’		\FB@xetex@punct@french: xeCJK.sty	

<code>\captionsfrench</code> to avoid a conflict with <code>papertex.cls</code> which loads <code>datetime.sty</code>	13	<code>\itemindent</code> . Suggested by Denis Bitouzé.	62
<code>\bbl@nonfrenchguillemets</code> deleted, use <code>\babel@save</code> instead.	32	<code>\extrasfrench</code> : Take advantage of babel's <code>\babel@savevariable</code> to handle apostrophe's <code>\lccode</code> . . .	15
Added explicit <code>\FBguillskip</code> for LuaTeX.	32	<code>\FBprocess@options</code> : Changed option <code>ThinColonSpace</code> to make it work also with LuaTeX.	55
Definitions of <code>\FB@og</code> and <code>\FB@fg</code> now depend on punctuation handling (LuaTeX / XeTeX / active). .	32	With koma-script and memoir class, customise <code>\captionformat</code> and <code>\captiondelim</code>	56
<code>french.cfg</code> will be loaded (if found) instead of <code>frenchb.cfg</code> . NO NEED for .cfg files in French anyway. .	66	<code>\FBthinskip</code> : LuaTeX requires dimensions: two new skips <code>\FBcolonskip</code> and <code>\FBthinskip</code> . .	17
In Plain, provide a substitute for <code>\PackageWarning</code> and <code>\PackageInfo</code>	14	<code>\frenchbsetup</code> : New options <code>OldFigTabCaptions</code> and <code>CustomiseFigTabCaptions</code>	47
Merging of <code>\captionsfrenchb</code> , <code>\captionsfrançais</code> with <code>\captionsfrench</code> deleted in favor of new babel 3.9 syntax.	43	v3.0b	
More informative, less TeXnical warning about <code>\@makecaption</code> . .	45	General: <code>frenchb.lua</code> was not found by Lua function <code>dofile</code> (not <code>kpathsea</code> aware). Call function <code>kpse.find_file</code> first, as suggested by Paul Gaborit.	24
New flag <code>\ifFB@luatex@punct</code> for 'high punctuation' management with LuaTeX engines.	16	Require <code>luatexbase</code> with LaTeX in case <code>fontspec</code> has not been loaded before babel.	17
New handling of 'high punctuation' through callbacks with LuaTeX engines.	17	v3.0c	
No warning about <code>\@makecaption</code> for SMF classes. No warning either with LuaTeX or XeTeX engines. .	44	General: Activate option <code>StandardLists</code> when beamer class is loaded. . .	48
Options processing completely reorganised.	47	Changed <code>\FBguill@spacing</code> (internal) to <code>\FBguillspace</code> (public). .	32
Support for options <code>frenchb</code> , <code>français</code> , <code>canadien</code> , <code>acadian</code> changed. .	13	<code>frenchb</code> requires babel-3.9i.	14
Test <code>\ifXeTeX</code> changed to <code>\ifBUnicode</code> and <code>'xltextra'</code> changed to <code>'fontspec'</code>	59	<code>frenchb.lua</code> : null glues should not trigger space insertion before high punctuation. Bug pointed out by Benoit Rivet for the 'lstlisting' environment of the listings package. .	20
<code>\CaptionSeparator</code> : Remove <code>\CaptionSeparatorORI</code> , use <code>\babel@save</code> instead.	43	Just load <code>luatexbase.sty</code> instead of <code>luaotfload.sty</code> with plain formats. .	17
<code>\captionsfrench</code> : Take advantage of babel's <code>\SetString</code> commands for <code>captionnames</code>	41	No need to define <code>\l@french</code> as <code>\lang@french</code> , babel.def (3.9j) takes care for this.	13
<code>\datefrench</code> : Take advantage of babel's <code>\SetString</code> commands for <code>\datefrench</code> . Doesn't work with Plain (yet?).	35	<code>\datefrench</code> : <code>\SetString</code> still does not work for Plain with babel 3.9k. Need to define <code>\datefrench</code> . . .	35
<code>\descriptionFB</code> : Add <code>\listindentFB</code>		<code>\frenchbsetup</code> : New option <code>INGuillSpace</code>	47
		v3.1a	
		General: Codes "13 and "14 added for French quotes in T1-encoding. Support for older versions of LuaTeX	

and XeTeX dropped.	51	of SmallCapsFigTabCaptions. Pointed out by Céline Chevalier. .	47
fontspec is not required for T1 fonts used with the luainputenc.sty package.	59	v3.1f General: \FBCaption@Separator changed when option Customise- FigTabCaptions is set to false. . .	45
frenchb.lua: added flag addgl which must also be true when prev or next is not a char (i.e. kern0 in «\texttt{a}»).	22	\FBprocess@options: Bug fix for the beamer class: figure and table captions are now consistent with frenchb's documentation. Pointed out by Denis Bitouzé.	56
frenchb.lua: codes 0x13 and 0x14 added for French quotes in T1- encoding.	18	Definition of \captionformat and \captiondelim changed when op- tion CustomiseFigTabCaptions is set to false.	56
frenchb.lua: look ahead when next is a kern (i.e. in «\texttt{a} »). .	22	\FBthinspace: \FBthinspace is no longer a kern but a skip (frenchb adds a nobreak penalty before it). .	17
Misplaced \fi for plain formats. . .	17	v3.1g General: frenchb.lua: flag addgl set to false for '«' at the end of an \hbox or a paragraph or when followed by a null glue (i.e. springs). . . .	22
New command \frquote for imbed- ded or long French quotations. . .	33	frenchb.lua: flag addgl set to false for '»' at the beginning of an \hbox or a paragraph or a tabular 'l' and 'c' columns.	22
\frenchbsetup: New options Inner- GuillSingle, EveryParGuill and Ev- eryLineGuill to control \frquote. .	47	frenchb.lua: node HLIST added; node TEMP added for the first node of \hboxes.	19
v3.1b General: frenchb.lua: add a check for null fid in french_punctuation (Tikz \nullfont). Bug pointed out by Paul Gaborit.	20	Lua function french_punctuation is now inserted at the end of the "kerning" callback (no prior- ity) instead of "hpack_filter" and "pre_linebreak_filter".	24
\captionsfrench: Change \scshape to customisable \FBfigtabshape for \figurename and \tablename.	41	Use Babel defined loops \bbl@for instead of \@for borrowed from file ltcntrl.dtx (\@for is undefined in Plain).	25
\fprimo): Removed \lowercase from definitions of \FrenchEnumerate, ... \No and co: \up already does the conversion.	38	\captionsfrench: \partname's defi- nition depends now on flag Part- NameFull. No need to redefine it in \frenchbsetup.	41
\frenchbsetup: New option Small- CapsFigTabCaptions.	47	Bug fix for koma-scripts classes: a spurious dot was added by the \partformat command.	42
\ieres: Removed \lowercase from definitions of \ieme and co: \up already does the conversion. . . .	37	\frenchbsetup: PartNameFull now just sets the flag, nothing to add to \captionsfrench when false. . .	47
v3.1c General: frenchb.lua: Previous bug fix for null glues (v3.0c) did not work properly. Fixed now (I hope). Pointed out by Jacques André. . .	20	v3.1h General: french.cfg from e-french con-	
v3.1d General: New section: issue warnings if packages listings, numprint and natbib are loaded too early or too late vs babel.	46		
v3.1e \frenchbsetup: Corrected typo: SmallCapsFigTabCaptions instead			

