

# A Babel language definition file for French

Daniel FLIPO  
daniel.flipo@free.fr

## 1 The French language

The file `frenchb.dtx`<sup>1</sup>, 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 (1996/05/31) as part of `babel-3.6beta`.

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

This new version (2.x) has been designed to be used with L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> and PlainT<sub>E</sub>X formats only. L<sup>A</sup>T<sub>E</sub>X-2.09 is no longer supported. Changes between version 1.6 and v2.6g are listed in subsection 1.4 p. 7.

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 ‘double punctuation’ (: ; ! ?) in French, others concern the general layout (i.e. layout of lists, footnotes, indentation of first paragraphs of sections) and should apply to the whole document.

Starting with version 2.2, `frenchb` behaves differently according to `babel`’s *main language* defined as the *last* option<sup>2</sup> at `babel`’s loading. When French is not `babel`’s main language, `frenchb` no longer alters the global 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*<sup>3</sup>:

1. the first paragraph of each section is indented (L<sup>A</sup>T<sub>E</sub>X 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{}`;
3. vertical spacing in general L<sup>A</sup>T<sub>E</sub>X lists is shortened;
4. footnotes are displayed “à la française”.

---

<sup>1</sup>The file described in this section has version number v2.6g and was last revised on 2013/12/30.

<sup>2</sup>Its name is kept in `\bb1@main@language`.

<sup>3</sup>For each item, hooks are provided to reset standard L<sup>A</sup>T<sub>E</sub>X settings or to emulate the behavior of former versions of `frenchb` (see command `\frenchbsetup{}`, section 1.2).

Regarding local typography, the command `\selectlanguage{french}` switches to the French language<sup>4</sup>, with the following effects:

1. French hyphenation patterns are made active;
2. ‘double punctuation’ (: ; ! ?) is made active<sup>5</sup> for correct spacing in French;
3. `\today` prints the date in French;
4. the caption names are translated into French (L<sup>A</sup>T<sub>E</sub>X only). The separator following the table or figure number in captions is printed as ‘ – ’ in French instead of ‘: ’, this setting can be changed by redefining `\CaptionSeparator` for French and `\CaptionSeparatorORI` for other languages.
5. the space after `\dots` is removed in French.

Some commands are provided in `frenchb` to make typesetting easier:

1. French quotation marks can be entered using the commands `\og` and `\fg` which work in L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> and PlainT<sub>E</sub>X, their appearance depending on what is available to draw them; even if you use L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> and T1-encoding, you should refrain from entering them as `<<~French quotation marks~>>`: `\og` and `\fg` provide better horizontal spacing. `\og` and `\fg` can be used outside French, they typeset then English quotes “ and ”.
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<sup>es</sup>).
3. Family names should be typeset in small capitals and never be hyphenated, the macro `\bsc` (boxed small caps) does this, e.g., `Leslie~\bsc{Lamport}` will produce Leslie 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°, 2°, 3°, 4°. `\FrenchEnumerate{6}` prints 6°.
5. Abbreviations for “Numéro(s)” and “numéro(s)” (N° N<sup>os</sup> n° and n<sup>os</sup> ) 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 unbreakable 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<sub>E</sub>Xbook p. 134). The command `\DecimalMathComma` makes the comma

---

<sup>4</sup> `\selectlanguage{français}` and `\selectlanguage{frenchb}` are kept for backward compatibility but should no longer be used.

<sup>5</sup>Actually, they are active in the whole document, only their expansions differ in French and outside French.

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 `'1er juin'` (no need for a forced space after `1\ier`).

## 1.2 Customisation

Up to version 1.6, customisation of `frenchb` was achieved by entering commands in `frenchb.cfg`. Version 2.0 introduced a new command `\frenchbsetup{}` using the `keyval` syntax which should make it easier to choose among the many options available. The command `\frenchbsetup{}` is to appear in the preamble only (after loading `babel`). Usage of a `frenchb.cfg` file for customisation *should definitely be avoided*.

`\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 brackets, 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 replaces the former command `\StandardLayout`. It might be used to avoid conflicts with classes or packages which customise lists or footnotes.
- `GlobalLayoutFrench=false` [`true*`] can be used, when French is the main language, to emulate what prior versions of `frenchb` (pre-2.2) did: lists, and first paragraphs of sections will 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`). This option replaces the former command `\FrenchLayout`.
- `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 `list`. This option replaces the former command `\FrenchListSpacingfalse`.

- `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).
- `StandardItemizeEnv=true` [`false*`]; `frenchb` redefines the `itemize` environment to suppress any vertical space between items of `itemize` lists in French and customise leftmargins. 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 leftmargins 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 stops `frenchb` from changing the labels in `itemize` lists in French.
- `ItemLabels=\textendash, \textbullet, \ding{43}, ..., [\textemdash*]`; when `StandardItemLabels=false` (the default), this option enables to choose the label used in `itemize` lists for all levels. The next three options do the same but each one for one level only. Note that the example `\ding{43}` requires `\usepackage{pifont}`.
- `ItemLabeli=\textendash, \textbullet, \ding{43}, ..., [\textemdash*]`
- `ItemLabelii=\textendash, \textbullet, \ding{43}, ..., [\textemdash*]`
- `ItemLabeliii=\textendash, \textbullet, \ding{43}, ..., [\textemdash*]`
- `ItemLabeliv=\textendash, \textbullet, \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 for all four options `ReduceListSpacing=false`, `StandardItemizeEnv=true`, `StandardEnumerateEnv=true` and `StandardItemLabels=true`.
- `IndentFirst=false` [`true*`]; `frenchb` normally forces indentation of the first paragraph of sections. When this option is set to `false`, the first paragraph of will look the same in French and in English (not indented).
- `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). The former commands `\StandardFootnotes` and `\FrenchFootnotes` are still there, `\StandardFootnotes` can be useful 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 older documents. 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 add a `\thinspace` before ‘:’ ‘!’ ‘?’ and a normal (unbreakable) space before ‘;’ (this is recommended by the French Imprimerie nationale). This is convenient in most cases but can lead to addition of spurious spaces in URLs or in MS-DOS paths but only if they are not typed using `\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 type `\string; \string: \string! \string?` instead of `; : ! ?` in case an unwanted space is added by `frenchb`.

- `ThinColonSpace=true [false]` changes the normal (unbreakable) space added before the colon ‘:’ to a thin space, so that the same amount of space is added before any of the four double punctuation characters. The default setting is supported by the French Imprimerie nationale.
- `LowercaseSuperscripts=false [true]`; by default `frenchb` inhibits the upercasing 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 and SMF classes do so), you will get “Première partie I”, “Deuxième partie II” instead; when this occurs, this option should be set to `false`, part titles will then be printed as “Partie I”, “Partie II”.
- `SuppressWarning=true [false]`; when true `frenchb` issues no warnings if `\@makecaption` has been redefined or if the `bigfoot` package is in use.
- `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 requires `inputenc` to be loaded with the proper encoding, it works with 8-bits encodings (`latin1`, `latin9`, `ansinew`, `applemac`,...) and multi-byte encodings (`utf8` and `utf8x`).

### 1.3 Hyphenation checks

Once you have built your format, a good precaution would be to perform some basic tests about hyphenation in French. For L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> I suggest this:

- run 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 \’ev\’enement alg\’ebre}
\showhyphens{signal container événement algèbre}
\end{document}
```

- check the hyphenations proposed by T<sub>E</sub>X 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.

**Options’ order** – Please remember that options are read in the order they appear inside the `\frenchbsetup` command. Someone wishing that `frenchb` leaves the layout of lists and footnotes untouched but caring for indentation of first paragraph of sections could choose `\frenchbsetup{StandardLayout,IndentFirst}` and get the expected layout. Choosing `\frenchbsetup{IndentFirst,StandardLayout}` would not lead to the expected result: option `IndentFirst` would be overwritten by `StandardLayout`.

## 1.4 Changes

### What's new in version 2.0?

Here is the list of all changes:

- Support for L<sup>A</sup>T<sub>E</sub>X-2.09 and for L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> in compatibility mode has been dropped. This version is meant for L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> and Plain based formats (like `bplain`). L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> formats based on m<sup>L</sup>T<sub>E</sub>X 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 table 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 which has to be loaded *after* `babel` 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).

### What's new in version 2.1?

New command `\fup` to typeset better looking superscripts. Former command `\up` is now defined as `\fup`, an option `\frenchbsetup{FrenchSuperscripts=false}` is provided for backward compatibility. `\fup` was designed using ideas from Jacques André, Thierry Bouche and René Fritz, thanks to them!

### 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 2.3?

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 2.4?

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 2.5?

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`- and `XeTeX`-based engines.

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



## 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.

## 2 The code

### 2.1 Initial setup

While this file was read through the option **frenchb** we make it behave as if **french** was specified.

```
1 \def\CurrentOption{french}
```

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

```
2 \LdfInit\CurrentOption\datefrench
```

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

```
3 \newif\ifLaTeXe
4 \let\bbl@tempa\relax
5 \ifx\magnification\@undefined
6   \ifx\@compatibilitytrue\@undefined
7     \PackageError{frenchb.ldf}
8       {LaTeX-2.09 format is no longer supported.\MessageBreak
9       Aborting here}
10    {Please upgrade to LaTeX2e!}
11   \let\bbl@tempa\endinput
12   \else
13     \LaTeXettrue
14   \fi
15 \fi
16 \bbl@tempa
```

Check if hyphenation patterns for the French language have been loaded in `language.dat`; we allow for the names 'french', 'français', 'canadien' or 'acadian'. The latter two are both names used in Canada for variants of French that are in use in that country.

```
17 \ifx\l@french\@undefined
18   \ifx\l@français\@undefined
19     \ifx\l@canadien\@undefined
20       \ifx\l@acadian\@undefined
21         \@nopatterns{French}
```

```

22      \adddialect\l@french0
23      \else
24      \let\l@french\l@acadian
25      \fi
26      \else
27      \let\l@french\l@canadien
28      \fi
29      \else
30      \let\l@french\l@français
31      \fi
32 \fi

```

Now `\l@french` is always defined.

The internal name for the French language is `french`; `français` and `frenchb` are synonymous for `french`: first let both names use the same hyphenation patterns. Later we will have to set aliases for `\captionfrench`, `\datefrench`, `\extrasfrench` and `\noextrasfrench`. As French uses the standard values of `\lefthyphenmin` (2) and `\righthyphenmin` (3), no special setting is required here.

```

33 \ifx\l@français\@undefined
34   \let\l@français\l@french
35 \fi
36 \ifx\l@frenchb\@undefined
37   \let\l@frenchb\l@french
38 \fi

```

When this language definition file was loaded for one of the Canadian versions of French we need to make sure that a suitable hyphenation pattern register will be found by  $\text{\TeX}$ .

```

39 \ifx\l@canadien\@undefined
40   \let\l@canadien\l@french
41 \fi
42 \ifx\l@acadian\@undefined
43   \let\l@acadian\l@french
44 \fi

```

This language definition can be loaded for different variants of the French language. The ‘key’ `babel` macros are only defined once, using ‘`french`’ as the language name, but `frenchb` and `français` are synonymous.

```

45 \def\datefrançais{\datefrench}
46 \def\datefrenchb{\datefrench}
47 \def\extrasfrançais{\extrasfrench}
48 \def\extrasfrenchb{\extrasfrench}
49 \def\noextrasfrançais{\noextrasfrench}
50 \def\noextrasfrenchb{\noextrasfrench}

```

`\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.

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

```

51 \newif\ifFBunicode
52 \newif\ifBBLuaTeX
53 \newif\ifFBXeTeX
54 \begingroup\expandafter\expandafter\expandafter\endgroup
55 \expandafter\ifx\csname luatexversion\endcsname\relax
56 \else
57   \FBunicodetrue \BBLuaTeXtrue
58 \fi
59 \begingroup\expandafter\expandafter\expandafter\endgroup
60 \expandafter\ifx\csname XeTeXrevision\endcsname\relax
61 \else
62   \FBunicodetrue \FBXeTeXtrue
63 \fi

```

These `\lccode` changes will ensure correct hyphenation of words like *d’aventure*, *l’utopie*, with all TeX engines (XeTeX, LuaTeX, pdfTeX) using `hyph-fr.tex` patterns.

```

64 \@namedef{extras\CurrentOption}{%
65     \ifFBunicode \lccode'\='2019 \lccode"2019="2019
66     \else        \lccode'\='\'
67     \fi}
68 \@namedef{noextras\CurrentOption}{%
69     \ifFBunicode \lccode"2019=\z@\fi
70     \lccode'\='=\z@}

```

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

```

71 \addto\extrasfrench{\bbl@frenchspacing}
72 \addto\noextrasfrench{\bbl@nonfrenchspacing}

```

## 2.2 Punctuation

As long as no better solution is available, the ‘double punctuation’ characters (; ! ? and :) have to be made `\active` for an automatic control of the amount of space to insert before them. XeTeX provides an alternative to active characters and LuaTeX will hopefully do so as well in the (near?) future.

Before doing so, we have to save the standard definition of `\@makecaption` (which includes two ‘:’) to compare it later to its definition at the `\begin{document}`.

```

73 \long\def\STD@makecaption#1#2{%
74   \vskip\abovecaptionskip
75   \sbox\@tempboxa{#1: #2}%
76   \ifdim \wd\@tempboxa >\hsize
77     #1: #2\par
78   \else
79     \global \@minipagefalse
80     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
81   \fi
82   \vskip\belowcaptionskip}

```

According to the I.N. specifications, the ‘.’ requires a normal space before it, but some people prefer a `\thinspace` (just like the other three). We define `\FBcolonspace` to hold the required amount of space (user customisable). In case some users are not satisfied with `\thinspace`’s width, it is also customisable.

```
83 \newcommand*\FBcolonspace{\space}
84 \newcommand*\FBthinspace{\thinspace}
```

```
\ifF@active@punct Check the availability of \XeTeXinterchartokenstate and decide whether the
\ifF@xetex@punct ‘double punctuation’ characters (; ! ? and :) have to be made \active or not.
```

```
85 \newif\ifFB@active@punct \FB@active@puncttrue
86 \newif\ifFB@xetex@punct
87 \begingroup\expandafter\expandafter\expandafter\endgroup
88 \expandafter\ifx\csname XeTeXinterchartokenstate\endcsname\relax
89 \else
90 \FB@xetex@puncttrue\FB@active@punctfalse
91 \fi
```

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 2.13).

For every character used in French text-mode (except spaces), `\XeTeXcharclass` value must be 0. `\XeTeXcharclass` value for spaces is assumed to be 255. Otherwise, the spacing before the ‘double 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.

```
92 \newif\ifFBAutoSpaceGuill \FBAutoSpaceGuilltrue
93 \newif\ifFBguillo@addspace
94 \newif\ifFBguillf@addspace
95 \newif\ifFBog@addspace \FBog@addspacetrue
96 \newif\ifFBfg@addspace \FBfg@addspacetrue
97 \newcount\FB@interchartokenstateORI
98 \ifFB@xetex@punct
99 \ifLaTeXe
100 \PackageInfo{frenchb.ldf}{No need for active punctuation characters
101 \MessageBreak with this version of XeTeX! reported}
102 \fi
```

We will need the following code (borrowed from `zhsusefulmacros.sty`) for loops:

```
103 \ifundefined{for}{%
104 \def\@nnil{\@nil}%
105 \def\@empty{}%
106 \def\@forloop#1\@#2#3{%
107 \long\def\@for#1:=#2\do#3{%
108 \expandafter\def\expandafter\@fortmp\expandafter{#2}%
109 \ifx\@fortmp\@empty \else
110 \expandafter\@forloop#2,\@nil,\@nil\@#1{#3}\fi}%
111 \long\def\@forloop#1,#2,#3\@#4#5{\def#4{#1}\ifx #4\@nnil \else
```

```

112      #5\def#4{#2}\ifx #4\@nnil \else#5\@iforloop #3\@@#4{#5}\fi\fi}%
113 \long\def\@iforloop#1,#2\@@#3#4{\def#3{#1}\ifx #3\@nnil
114   \expandafter\@fornoop \else
115   #4\relax\expandafter\@iforloop\fi#2\@@#3{#4}}%
116 \def\@tfor#1:={\@tfor#1 }%
117 \long\def\@tfor#1#2\do#3{\def\@fortmp{#2}\ifx\@fortmp\space\else
118   \@tforloop#2\@nil\@nil\@@#1{#3}\fi}%
119 \long\def\@tforloop#1#2\@@#3#4{\def#3{#1}\ifx #3\@nnil
120   \expandafter\@fornoop \else
121   #4\relax\expandafter\@tforloop\fi#2\@@#3{#4}}%
122 }-}%

```

Five new character classes are defined for frenchb.

```

123 \newXeTeXintercharclass\FB@punctthick
124 \newXeTeXintercharclass\FB@punctthin
125 \newXeTeXintercharclass\FB@punctnul
126 \newXeTeXintercharclass\FB@punctguilo
127 \newXeTeXintercharclass\FB@punctguilf

```

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.

```

128 \def\FB@charclassesORI{}
129 \def\empty{}
130 \def\FB@parse#1,#2\endparse{\def\FB@class{#1}%
131   \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.

```

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

```

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

```

133 \ifx\FB@charclassesORI\empty
134   \FB@interchartokenstateORI=\XeTeXinterchartokenstate
135   \@for\FB@char:={\:,\;,\!,\?, "AB,"BB,%
136     \(\, \[, \{, \., \-, \), \], \},%
137     \%, "22,"27,"60,"2019,"A0,"202F}\do
138     {\edef\FB@charclassesORI{\FB@charclassesORI%
139       \the\XeTeXcharclass\FB@char,}}%
140   \let\shorthandonORI\shorthandon
141   \let\shorthandoffORI\shorthandoff
142 \fi

```

Set the classes and interactions between classes.

```

143 \XeTeXinterchartokenstate=1
144 \XeTeXcharclass \: = \FB@punctthick
145 \XeTeXinterchartoks \z@ \FB@punctthick = {%
146   \ifhmode\FDP@colonspace\fi}%
147 \XeTeXinterchartoks \FB@punctguilf \FB@punctthick = {%
148   \FDP@colonspace}%
149 \XeTeXinterchartoks 255 \FB@punctthick = {%
150   \ifhmode\unskip\penalty\@M\FB@colonspace\fi}%
151 \@for\FB@char:={\:,\;,\!,\?}\do
152   {\XeTeXcharclass\FB@char=\FB@punctthin}%
153 \XeTeXinterchartoks \z@ \FB@punctthin = {%

```



```

196     \fi
197   }
198   \addto\extrasfrench{\FB@xetex@punct@french}
199   \addto\noextrasfrench{\FB@xetex@punct@nonfrench}
200 \fi

```

Otherwise we need to make the four characters ; ! ? and : ‘active’ and provide their definitions.

```

201 \ifFB@active@punct
202   \initiate@active@char{:}%
203   \initiate@active@char{;}%
204   \initiate@active@char{!}%
205   \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 `\thinspace` instead. If no space has been typed, we add `\FDP@thinspace` which will be defined, up to the user’s wishes, as an automatic added thin space, or as `\@empty`.

```

206   \declare@shorthand{french}{;}{%
207     \ifhmode
208       \ifdim\lastskip>\z@
209         \unskip\penalty\@M\FBthinspace
210       \else
211         \FDP@thinspace
212       \fi
213     }

```

Now we can insert a ; character.

```

214     \string;}

```

The next three definitions are very similar.

```

215   \declare@shorthand{french}{!}{%
216     \ifhmode
217       \ifdim\lastskip>\z@
218         \unskip\penalty\@M\FBthinspace
219       \else
220         \FDP@thinspace
221       \fi
222     \fi
223     \string!}
224   \declare@shorthand{french}{?}{%
225     \ifhmode
226       \ifdim\lastskip>\z@
227         \unskip\penalty\@M\FBthinspace
228       \else
229         \FDP@thinspace
230       \fi
231     \fi
232     \string?}
233   \declare@shorthand{french}{:}{%
234     \ifhmode
235       \ifdim\lastskip>\z@
236         \unskip\penalty\@M\FBcolonspace

```

```

237         \else
238         \FDP@colonspace
239         \fi
240     \fi
241     \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.

```

242 \declare@shorthand{system}{:}{\string:}
243 \declare@shorthand{system}{!}{\string!}
244 \declare@shorthand{system}{?}{\string?}
245 \declare@shorthand{system}{;}{\string;}
246 %}

```

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

```

247 \addto\extrasfrench{%
248     \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.

```

249     \bbl@activate{:}\bbl@activate{;}%
250     \bbl@activate{!}\bbl@activate{?}%
251 }
252 \addto\noextrasfrench{%
253     \bbl@deactivate{:}\bbl@deactivate{;}%
254     \bbl@deactivate{!}\bbl@deactivate{?}}
255 \fi

```

A new ‘if’ `\FBAutoSpacePunctuation` needs to be defined now.

```

256 \newif\iffBAutoSpacePunctuation \FBAutoSpacePunctuationtrue

```

`\AutoSpaceBeforeFDP` and `\FDP@thinspace` and `\FDP@colonspace` are defined as unbreakable spaces by `\autospace@beforeFDP` or as `\@empty` by `\noautospace@beforeFDP` (internal commands), user commands `\AutoSpaceBeforeFDP` and `\NoAutoSpaceBeforeFDP` do the same and take care of the flag `\iffBAutoSpacePunctuation` in L<sup>A</sup>T<sub>E</sub>X. Set the default now for Plain (done later for L<sup>A</sup>T<sub>E</sub>X).

```

257 \def\autospace@beforeFDP{%
258     \def\FDP@thinspace{\penalty\@M\FBthinspace}%
259     \def\FDP@colonspace{\penalty\@M\FBcolonspace}}
260 \def\noautospace@beforeFDP{\let\FDP@thinspace\@empty
261                             \let\FDP@colonspace\@empty}
262 \ifLaTeXe
263     \def\AutoSpaceBeforeFDP{\autospace@beforeFDP
264                             \FBAutoSpacePunctuationtrue}
265     \def\NoAutoSpaceBeforeFDP{\noautospace@beforeFDP
266                             \FBAutoSpacePunctuationfalse}
267 \else
268     \let\AutoSpaceBeforeFDP\autospace@beforeFDP
269     \let\NoAutoSpaceBeforeFDP\noautospace@beforeFDP
270 \fi
271 \AutoSpaceBeforeFDP

```



In L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> `\ttfamily` (and hence `\texttt`) will be redefined ‘AtBeginDocument’ as `\ttfamilyFB` so that no space is added before the four ; : ! ? characters, even if `AutoSpacePunctuation` is true. `\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.

```

272 \ifLaTeXe
273   \let\ttfamilyORI\ttfamily
274   \let\rmfamilyORI\rmfamily
275   \let\sffamilyORI\sffamily
276   \DeclareRobustCommand\ttfamilyFB{%
277     \FBAutoSpaceGuillfalse
278     \FBog@addspacetrue      \FBfg@addspacetrue
279     \noautospace@beforeFDP\ttfamilyORI}%
280   \DeclareRobustCommand\rmfamilyFB{%
281     \FBAutoSpaceGuilltrue
282     \ifFBguillo@addspace\FBog@addspacefalse\fi
283     \ifFBguillf@addspace\FBfg@addspacefalse\fi
284     \ifFBAutoSpacePunctuation
285       \autospace@beforeFDP
286     \else
287       \noautospace@beforeFDP
288     \fi
289     \rmfamilyORI}%
290   \DeclareRobustCommand\sffamilyFB{%
291     \FBAutoSpaceGuilltrue
292     \ifFBguillo@addspace\FBog@addspacefalse\fi
293     \ifFBguillf@addspace\FBfg@addspacefalse\fi
294     \ifFBAutoSpacePunctuation
295       \autospace@beforeFDP
296     \else
297       \noautospace@beforeFDP
298     \fi
299     \sffamilyORI}%
300 \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 T<sub>E</sub>X and X<sub>E</sub>T<sub>E</sub>X based engines) and is meant to be used inside a group.

```

301 \newcommand*{\NoAutoSpacing}{\FBAutoSpaceGuillfalse
302   \FBog@addspacetrue      \FBfg@addspacetrue
303   \ifFB@active@punct\shorthandoff{;:!?}\fi
304   \ifFB@xetex@punct\XeTeXinterchartokenstate=0\fi
305 }

```

## 2.3 Commands for French quotation marks

`\og` The top macros for quotation marks will be called `\og` (“ouvrez guillemets”) and  
`\fg` `\fg` (“fermez guillemets”). Another option for typesetting quotes in multilingual  
texts is to use the package `csquotes.sty` and its command `\enquote`.

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

`\guillemotleft` L<sup>A</sup>T<sub>E</sub>X users are supposed to use 8-bit output encodings (T1, LY1,...) to typeset  
`\guillemotright` French, those who still stick to OT1 should call `aeguill.sty` or a similar package.  
`\textquotedblleft` In both cases the commands `\guillemotleft` and `\guillemotright` will print the  
`\textquotedblright` French opening and closing quote characters from the output font. For XeLaTeX,  
`\guillemotleft` and `\guillemotright` are defined by package `xunicode.sty`.  
We will check ‘AtBeginDocument’ that the proper output encodings are in use  
(see end of section 2.13).

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

```
308 \ifLaTeXe
309 \else
310   \ifFBunicode
311     \def\guillemotleft{{\char"00AB}}
312     \def\guillemotright{{\char"00BB}}
313     \def\textquotedblleft{{\char"201C}}
314     \def\textquotedblright{{\char"201D}}
315   \else
316     \def\guillemotleft{\leavevmode\raise0.25ex
317                           \hbox{$\scriptscriptstyle\ll$}}
318     \def\guillemotright{\raise0.25ex
319                           \hbox{$\scriptscriptstyle\gg$}}
320     \def\textquotedblleft{‘}
321     \def\textquotedblright{’}
322   \fi
323   \let\xspace\relax
324 \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.  
`\FBguill@spacing` which does the spacing, has been fine tuned by Thierry  
Bouche. 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).

```
325 \newcommand*\FBguill@spacing{\penalty\@M\hskip.8\fontdimen2\font
326                               plus.3\fontdimen3\font
327                               minus.8\fontdimen4\font}
328 \DeclareRobustCommand*\FB@og{\leavevmode\guillemotleft
329                               \ifFBog@addspace\FBguill@spacing\fi}
330 \DeclareRobustCommand*\FB@fg{\ifdim\lastskip>\z@\unskip\fi
331                               \ifFBfg@addspace\FBguill@spacing\fi
332                               \guillemotright\xspace}
```

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.

```

333 \ifLaTeXe
334   \def\bbl@frenchguillemets{\renewcommand*{\og}{\FB@og}%
335                               \renewcommand*{\fg}{\FB@fg}}
336   \def\bbl@nonfrenchguillemets{\renewcommand*{\og}{\textquotedblleft}%
337                                 \renewcommand*{\fg}{\ifdim\lastskip>\z@ \unskip\fi
338                                     \textquotedblright}}
339 \else
340   \def\bbl@frenchguillemets{\let\og\FB@og
341                               \let\fg\FB@fg}
342   \def\bbl@nonfrenchguillemets{\def\og{\textquotedblleft}%
343                                 \def\fg{\ifdim\lastskip>\z@ \unskip\fi \textquotedblright}}
344 \fi
345 \addto\extrasfrench{\bbl@frenchguillemets}
346 \addto\noextrasfrench{\bbl@nonfrenchguillemets}

```

## 2.4 Date in French

`\datefrench` The macro `\datefrench` redefines the command `\today` to produce French dates.

```

347 \@namedef{date\CurrentOption}{%
348   \def\today{\number\day\ifnum1=\day {\ier}\fi \space
349     \ifcase\month
350       \or janvier\or f{\FB@eacute}vrier\or mars\or avril\or mai\or
351       juin\or juillet\or ao{\FB@ucirconflexe}t\or septembre\or
352       octobre\or novembre\or d{\FB@eacute}cembre\fi
353   \space \number\year}}

```

## 2.5 Extra utilities

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

`\up` `\up` eases the typesetting of superscripts like '1<sup>er</sup>'. Up to version 2.0 of `frenchb`  
`\fup` `\up` was just a shortcut for `\textsuperscript` in  $\text{\LaTeX} 2_{\epsilon}$ , but several users complained that `\textsuperscript` typesets superscripts too high and too big, so we now define `\fup` as an attempt to produce better looking superscripts. `\up` is defined as `\fup` but `\frenchbsetup{FrenchSuperscripts=false}` redefines `\up` as `\textsuperscript` for compatibility with previous versions.

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

```

354 \newif\ifFB@poorman
355 \newdimen\FB@Mht
356 \ifLaTeXe
357   \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{}`.

```

358 \newcommand*\FBsupR{-0.12}
359 \newcommand*\FBsupS{0.65}
360 \newcommand*\FB@lc[1]{\MakeLowercase{#1}}
361 \DeclareRobustCommand*\FB@up@fake[1]{%
362   \settoheight{\FB@Mht}{M}%
363   \addtolength{\FB@Mht}{\FBsupR \FB@Mht}%
364   \addtolength{\FB@Mht}{-\FBsupS ex}%
365   \raisebox{\FB@Mht}{\scalefont{\FBsupS}{\FB@lc{#1}}}%
366 }

```

The only packages I currently know to take advantage of real superscripts are a) `xltxtra` used in conjunction with XeLaTeX and OpenType fonts having the font feature 'VerticalPosition=Superior' (`xltxtra` defines `\realsuperscript` and `\fakesuperscript`) 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.

```

367 \def\FB@split#1#2#3#4\@nil{\def\FB@firstthree{#1#2#3}%
368   \def\FB@suffix{#4}}
369 \def\FB@x{x}
370 \def\FB@j{j}
371 \DeclareRobustCommand*\FB@up[1]{%
372   \bgroup \FB@poormantrue
373   \expandafter\FB@split\f@family\@nil

```

Then `\FB@up` looks for a .fd file named `t1fut-sup.fd` (Fourier) or `t1ppl-sup.fd` (Palatino), etc. supposed to define the subfamily (`fut-sup` or `ppl-sup`, etc.) giving access to the built-in superscripts. If the .fd file is not found by `\IfFileExists`, `\FB@up` falls back on fake superscripts, otherwise `\FB@suffix` is checked to decide whether to use fake or real superscripts.

```

374 \edef\reserved@a{\lowercase{%
375   \noexpand\IfFileExists{\f@encoding\FB@firstthree -sup.fd}}}%
376 \reserved@a
377 {\ifx\FB@suffix\FB@x \FB@poormanfalse\fi
378   \ifx\FB@suffix\FB@j \FB@poormanfalse\fi
379   \ifFB@poorman \FB@up@fake{#1}%
380   \else \FB@up@real{#1}%
381   \fi}%
382 {\FB@up@fake{#1}}%
383 \egroup

```

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

```
384 \newcommand*{\FB@up@real}[1]{\bgroup
385     \fontfamily{\FB@firstthree -sup}\selectfont \FB@lc{#1}\egroup}
\fbp is now defined as \FB@up unless \realsuperscript is defined (occurs with
XeLaTeX calling xltextra.sty).
```

```
386 \DeclareRobustCommand*{\fbp}[1]{%
387     \@ifundefined{realsuperscript}%
388     {\FB@up{#1}\let\realsuperscript\undefined}%
389     {\bgroup\let\fakesuperscript\FB@up@fake
390         \realsuperscript{\FB@lc{#1}}\egroup}}
```

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

```
391 \providecommand*{\up}{\relax}
Poor man's definition of \up for Plain.
392 \else
393 \providecommand*{\up}[1]{\leavevmode\raise1ex\hbox{\sevenrm #1}}
394 \fi
```

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

```
\ier 395 \def\ieme{\up{\lowercase{e}}\xspace}
\iere 396 \def\iemes{\up{\lowercase{es}}\xspace}
\iemes 397 \def\ier{\up{\lowercase{er}}\xspace}
\iers 398 \def\iers{\up{\lowercase{ers}}\xspace}
\ieres 399 \def\iere{\up{\lowercase{re}}\xspace}
400 \def\ieres{\up{\lowercase{res}}\xspace}
```

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

```
\no 401 \newcommand*{\FrenchEnumerate}[1]{%
\nos 402     #1\up{\lowercase{o}}\kern+.3em}
\nos 403 \newcommand*{\FrenchPopularEnumerate}[1]{%
\primo 404     #1\up{\lowercase{o}}\kern+.3em}
```

`\fprimo`) Typing `\primo` should result in '1°',

```
405 \def\primo{\FrenchEnumerate1}
406 \def\secundo{\FrenchEnumerate2}
407 \def\tertio{\FrenchEnumerate3}
408 \def\quarto{\FrenchEnumerate4}
while typing \fprimo) gives '1°'.
409 \def\fprimo{\FrenchPopularEnumerate1}
410 \def\fsecundo{\FrenchPopularEnumerate2}
411 \def\ftertio{\FrenchPopularEnumerate3}
412 \def\fquarto{\FrenchPopularEnumerate4}
```

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

```
413 \DeclareRobustCommand*{\No}{N\up{\lowercase{o}}\kern+.2em}
414 \DeclareRobustCommand*{\no}{n\up{\lowercase{o}}\kern+.2em}
415 \DeclareRobustCommand*{\Nos}{N\up{\lowercase{os}}\kern+.2em}
416 \DeclareRobustCommand*{\nos}{n\up{\lowercase{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}`.

```
417 \DeclareRobustCommand*\bsc}[1]{\leavevmode\begingroup\kern0pt
418                               \scshape #1\endgroup}
419 \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.

```
420 \ifFBunicode
421   \newcommand*\at{{\char"0040}}
422   \newcommand*\circonflexe{{\char"005E}}
423   \newcommand*\tild{{\char"007E}}
424   \newcommand*\boi{{\textbackslash}}
425   \newcommand*\degre{{\char"00B0}}
426 \else
427   \ifLaTeXe
428     \DeclareTextSymbol{\at}{T1}{64}
429     \DeclareTextSymbol{\circonflexe}{T1}{94}
430     \DeclareTextSymbol{\tild}{T1}{126}
431     \DeclareTextSymbolDefault{\at}{T1}
432     \DeclareTextSymbolDefault{\circonflexe}{T1}
433     \DeclareTextSymbolDefault{\tild}{T1}
434     \DeclareRobustCommand*\boi{{\textbackslash}}
435     \DeclareRobustCommand*\degre{{\r{}}}
436   \else
437     \def\T@one{T1}
438     \ifx\f@encoding\T@one
439       \newcommand*\degre{{\char6}}
440     \else
441       \newcommand*\degre{{\char23}}
442     \fi
443     \newcommand*\at{{\char64}}
444     \newcommand*\circonflexe{{\char94}}
445     \newcommand*\tild{{\char126}}
446     \newcommand*\boi{{\backslash}}
447   \fi
448 \fi
```

French dates and captions make use of four non-ascii characters (à, è, é and û). This is fine except for (plain) XeTeX (`\accent` commands are not implemented), so we define four new commands to deal with this issue.

```
449 \newcommand*\FBgrave{{\`a}}
450 \newcommand*\FBgrave{{\`e}}
451 \newcommand*\FBacute{{\`e}}
452 \newcommand*\FBucirconflexe{{\^u}}
453 \ifFBunicode
454   \ifLaTeXe
```

```

455 \else
456   \def\FBgrave{{\char"00E0}}
457   \def\FBgrave{{\char"00E8}}
458   \def\FBeacute{{\char"00E9}}
459   \def\FBucirconflexe{{\char"00FB}}
460 \fi
461 \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 0.3em, this lets the symbol ‘degree’ stick to the preceding (e.g., `45\degrees`) or following character (e.g., `20~\degrees C`). If T<sub>E</sub>X 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.

```

462 \ifLaTeXe
463   \newcommand*{\degrees}{\degree}
464   \ifFBunicode
465     \DeclareRobustCommand*{\degrees}{\degree}
466   \else
467     \def\Warning@degree@TSone{%
468       \PackageWarning{frenchb.ldf}{%
469         Degrees would look better in TS1-encoding:
470         \MessageBreak add \protect
471         \usepackage{textcomp} to the preamble.
472         \MessageBreak Degrees used}}
473     \AtBeginDocument{\@ifundefined{DeclareEncodingSubset}%
474       {\DeclareRobustCommand*{\degrees}{%
475         \leavevmode\hbox to 0.3em{\hss\degree\hss}%
476         \Warning@degree@TSone
477         \global\let\Warning@degree@TSone\relax}}%
478       {\DeclareRobustCommand*{\degrees}{%
479         \hbox{\UseTextSymbol{TS1}{\textdegree}}}%
480       }
481   \fi
482 \else
483   \newcommand*{\degrees}{%
484     \leavevmode\hbox to 0.3em{\hss\degree\hss}}
485 \fi

```

## 2.6 Formatting numbers

**\DecimalMathComma** As mentioned in the T<sub>E</sub>Xbook 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.

```

486 \newcount\std@mcc
487 \newcount\dec@mcc
488 \std@mcc=\mathcode'\,

```

```

489 \dec@mcc=\std@mcc
490 \@tempcnta=\std@mcc
491 \divide\@tempcnta by "1000
492 \multiply\@tempcnta by "1000
493 \advance\dec@mcc by -\@tempcnta
494 \newcommand*\DecimalMathComma{\iflanguage{french}%
495     {\mathcode'\,=\dec@mcc}{}}%
496     \addto\extrasfrench{\mathcode'\,=\dec@mcc}}
497 \newcommand*\StandardMathComma{\mathcode'\,=\std@mcc
498     \addto\extrasfrench{\mathcode'\,=\std@mcc}}
499 \addto\noextrasfrench{\mathcode'\,=\std@mcc}

```

`\nombre` The command `\nombre` is now borrowed from `numprint.sty` for L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>. 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. of the change.

```

500 \newcommand*\nombre[1]{\#1}\message{%
501     *** \noexpand\nombre no longer formats numbers\string! ***}}%

```

The next definitions only make sense for L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>. Let's cleanup and exit if the format in Plain based.

```

502 \let\FBstop@here\relax
503 \def\FBclean@on@exit{\let\ifLaTeXe\undefined
504     \let\LaTeXetrue\undefined
505     \let\LaTeXefalse\undefined}
506 \ifx\magnification\@undefined
507 \else
508     \def\FBstop@here{\let\STD@makecaption\relax
509         \FBclean@on@exit
510         \ldf@quit\CurrentOption\endinput}
511 \fi
512 \FBstop@here

```

What follows now is for L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> *only*. We redefine `\nombre` for L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>. 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.

```

513 \renewcommand*\nombre[1]{\Warning@nombre\numprint{#1}}
514 \newcommand*\Warning@nombre{%
515     \ifundefined{numprint}%
516         {\PackageWarning{frenchb.1df}{%
517             \protect\nombre\space now relies on package numprint.sty,
518             \MessageBreak add \protect
519             \usepackage[autolanguage]{numprint}\MessageBreak
520             to your preamble *after* loading babel, \MessageBreak
521             see file numprint.pdf for more options.\MessageBreak
522             \protect\nombre\space called}%
523         \global\let\Warning@nombre\relax
524         \global\let\numprint\undefined
525     }{}%
526 }

```



```

527 \newcommand*{\ThinSpaceInFrenchNumbers}{%
528   \PackageWarning{frenchb.1df}{%
529     Type \protect\frenchbsetup{\ThinSpaceInFrenchNumbers}
530     \MessageBreak Command \protect\ThinSpaceInFrenchNumbers\space
531     is no longer\MessageBreak defined in frenchb v.~2,}}

```

## 2.7 Caption names

The next step consists of defining the French equivalents for the L<sup>A</sup>T<sub>E</sub>X caption names.

`\captionsfrench` Let's first define `\captionsfrench` which sets all strings used in the four standard document classes provided with L<sup>A</sup>T<sub>E</sub>X.

```

532 \@namedef{captions\CurrentOption}{%
533   \def\refname{R{\FBacute}f{\FBacute}rences}%
534   \def\abstractname{R{\FBacute}sum{\FBacute}}%
535   \def\bibname{Bibliographie}%
536   \def\prefacename{Pr{\FBacute}face}%
537   \def\chaptername{Chapitre}%
538   \def\appendixname{Annexe}%
539   \def\contentsname{Table des mati{\FBgrave}res}%
540   \def\listfigurename{Table des figures}%
541   \def\listtablename{Liste des tableaux}%
542   \def\indexname{Index}%
543   \def\figurename{{\scshape Figure}}%
544   \def\tablename{{\scshape Table}}%

  "Première partie" instead of "Part I".

545   \def\partname{\protect\@Fpt partie}%
546   \def\@Fpt{{\ifcase\value{part}\or Premi{\FBgrave}re\or
547     Deuxi{\FBgrave}me\or Troisi{\FBgrave}me\or
548     Quatri{\FBgrave}me\or Cinqui{\FBgrave}me\or
549     Sixi{\FBgrave}me\or Septi{\FBgrave}me\or Huiti{\FBgrave}me\or
550     Neuvi{\FBgrave}me\or Dixi{\FBgrave}me\or Onzi{\FBgrave}me\or
551     Douzi{\FBgrave}me\or Treizi{\FBgrave}me\or
552     Quatorzi{\FBgrave}me\or Quinzi{\FBgrave}me\or
553     Seizi{\FBgrave}me\or Dix-septi{\FBgrave}me\or
554     Dix-huiti{\FBgrave}me\or Dix-neuvi{\FBgrave}me\or
555     Vingti{\FBgrave}me\fi}\space\def\thepart{}}%
556   \def\pagename{page}%
557   \def\seename{voir}%
558   \def\alsoname{voir aussi}%
559   \def\enclname{P.~J. }%
560   \def\ccname{Copie {\FBgrave} }%
561   \def\headtoname{}%
562   \def\proofname{D{\FBacute}monstration}%
563   \def\glossaryname{Glossaire}%
564   }

```

As some users who choose `frenchb` or `francais` as option of `babel`, might customise `\captionsfrenchb` or `\captionsfrancais` in the preamble, we merge their changes at the `\begin{document}` when they do so. The other variants of French

(canadien, acadian) are defined by checking if the relevant option was used and then adding one extra level of expansion.

```

565 \AtBeginDocument{\let\captions@French\captionsfrench
566                 \@ifundefined{captionsfrenchb}%
567                 {\let\captions@Frenchb\relax}%
568                 {\let\captions@Frenchb\captionsfrenchb}%
569                 \@ifundefined{captionsfrançais}%
570                 {\let\captions@Français\relax}%
571                 {\let\captions@Français\captionsfrançais}%
572                 \def\captionsfrench{\captions@French
573                 \captions@Français\captions@Frenchb}%
574                 \def\captionsfrançais{\captionsfrench}%
575                 \def\captionsfrenchb{\captionsfrench}%
576                 }
577 \@ifpackagewith{babel}{canadien}{%
578   \def\captionscanadien{\captionsfrench}%
579   \def\datecanadien{\datefrench}%
580   \def\extrascanadien{\extrasfrench}%
581   \def\noextrascanadien{\noextrasfrench}%
582 }{}
583 \@ifpackagewith{babel}{acadian}{%
584   \def\captionacadian{\captionsfrench}%
585   \def\dateacadian{\datefrench}%
586   \def\extrasacadian{\extrasfrench}%
587   \def\noextrasacadian{\noextrasfrench}%
588 }{}

```

`\CaptionSeparator`  
`\CaptionSeparatorORI`

Let's consider now captions in figures and tables. In French, captions in figures and tables should be printed with the number followed by endash (‘ – ’), never as ‘Figure 1:’ which is the default in standard L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> classes.

The standard definition of `\@makecaption` (e.g., the one provided in `article.cls`, `report.cls`, `book.cls` which is frozen for L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> according to Frank Mittelbach), has been saved in `\STD@makecaption` before making ‘:’ active (see section 2.2). ‘AtBeginDocument’ we compare it to its current definition (some classes like 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 `\CaptionSeparatorORI` (‘: ’) as in the standard `\@makecaption`, and will be changed to `\CaptionSeparator` (‘ – ’) in French. 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 AMS classes as their layout of captions is compatible with French typographic standards.

With koma-script classes, `frenchb` customises `\captionformat` in French and issues no warning.

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

```

589 \newif\if@FBwarning@capsep
590 \@FBwarning@capseptrue
591 \newcommand{\FBWarning}[2]{\PackageWarning{#1}{#2}}
592 \newcommand*\{\CaptionSeparatorORI}{\string\space}

```

```

593 \newcommand*{\CaptionSeparator}{\space\textendash\space}
594 \def\FBCaption@Separator{\CaptionSeparatorORI}
595 \long\def\FB@makecaption#1#2{%
596   \vskip\abovcaptionskip
597   \sbox\@tempboxa{#1\FBCaption@Separator #2}%
598   \ifdim \wd\@tempboxa >\hsize
599     #1\FBCaption@Separator #2\par
600   \else
601     \global \@minipagefalse
602     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
603   \fi
604   \vskip\belowcaptionskip}

```

Disable the standard warning with AMS classes.

```

605 \@ifclassloaded{amsart}{\@FBwarning@capsepfalse}{}
606 \@ifclassloaded{amsbook}{\@FBwarning@capsepfalse}{}
607 \@ifclassloaded{amstex}{\@FBwarning@capsepfalse}{}
608 \@ifclassloaded{amslatex}{\@FBwarning@capsepfalse}{}
609 \@ifclassloaded{amproc}{\@FBwarning@capsepfalse}{}

```

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

```

610 \AtBeginDocument{%
611   \@ifundefined{captionformat}%
612     {\let\captionformat\undefined}%
613     {\@FBwarning@capsepfalse}%
614 }

```

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.

```

615 \@ifpackageloaded{caption}
616   {\FBWarning{frenchb.ldb}%
617     {Please load the "caption" package\MessageBreak
618       AFTER babel/frenchb; reported}%
619   \@FBwarning@capsepfalse}%
620   {}

```

Same for package `floatrow`.

```

621 \@ifpackageloaded{floatrow}
622   {\FBWarning{frenchb.ldb}%
623     {Please load the "floatrow" package\MessageBreak
624       AFTER babel/frenchb; reported}%
625   \@FBwarning@capsepfalse}%
626   {}

```

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`.

```

627 \AtBeginDocument{%
628   \ifx\@makecaption\STD@makecaption
629     \global\let\@makecaption\FB@makecaption
630     \@FBwarning@capsepfalse
631   \fi
632   \if@FBwarning@capsep
633     \FBWarning{frenchb.ldb}%

```

```

634      {The definition of \protect\@makecaption\space
635       has been changed,\MessageBreak
636       frenchb will NOT customise it;\MessageBreak reported}%
637   \fi
638   \let\FB@makecaption\relax
639   \let\STD@makecaption\relax
640 }

```

In French, `\FB@makecaption` uses `\CaptionSeparator` which defaults to ‘–’ and `\CaptionSeparatorORI` (default ‘:’) for other languages.

```

641 \addto\extrasfrench{%
642   \def\FBCaption@Separator{\CaptionSeparator}}
643 \addto\noextrasfrench{%
644   \def\FBCaption@Separator{\CaptionSeparatorORI}}

```

## 2.8 French lists

`\listFB` Vertical spacing in lists should be shorter in French texts than the defaults provided by L<sup>A</sup>T<sub>E</sub>X. Note that the easy way, just changing values of vertical spacing parameters when entering French and restoring them to their defaults on exit would not work; so we define the command `\FB@listVsettings` to hold the settings to be used by the French variant `\listFB` of `\list`. Note that switching to `\listFB` reduces vertical spacing in *all* environments built on `\list`: `itemize`, `enumerate`, `description`, but also `abstract`, `quotation`, `quote` and `verse`... The amount of vertical space before and after a list is given by `\topsep` + `\parskip` (+ `\partopsep` if the list starts a new paragraph). IMHO, `\parskip` should be added *only* when the list starts a new paragraph, so I subtract `\parskip` from `\topsep` and add it back to `\partopsep`; this will normally make no difference because `\parskip`’s default value is 0pt, but will be noticeable when `\parskip` is *not* null.

`\endlist` is not redefined, but `\endlistORI` and `\endlistFB` will be defined as `\endlist` ‘AtBeginDocument’ for users who define their own lists using `\listORI` or `\listFB`.

```

645 \let\listORI\list
646 \def\FB@listVsettings{%
647   \setlength{\itemsep}{0.4ex plus 0.2ex minus 0.2ex}%
648   \setlength{\parsep}{0.4ex plus 0.2ex minus 0.2ex}%
649   \setlength{\topsep}{0.8ex plus 0.4ex minus 0.4ex}%
650   \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`.

```

651   \@tempdima=\parskip
652   \addtolength{\topsep}{-\@tempdima}%
653   \addtolength{\partopsep}{\@tempdima}%
654 }
655 \def\listFB#1#2{\listORI{#1}{\FB@listVsettings #2}}

```

Let’s now consider French itemize-lists. They differ from those provided by the standard L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> classes:

- The ‘•’ is never used in French itemize-lists, an emdash ‘—’ or an endash ‘–’ is preferred for all levels. The item label to be used in French is stored in `\FrenchLabelItem`, it defaults to ‘—’ and can be changed using `\frenchbsetup{}` (see section 2.13).
- Vertical spacing between items, before and after the list, should be *null* with *no glue* added;
- In French the labels of itemize-lists are vertically aligned as follows:

Text starting at ‘parindent’  
 ⇐ Leftmargin  
 – first item...  
   – first second level item  
   – next one...  
 – second item...

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

```
\Frlabelitemi 656 \newcommand*\FrenchLabelItem{\textemdash}
\Frlabelitemii 657 \newcommand*\Frlabelitemi{\FrenchLabelItem}
\Frlabelitemiii 658 \newcommand*\Frlabelitemii{\FrenchLabelItem}
\Frlabelitemiv 659 \newcommand*\Frlabelitemiii{\FrenchLabelItem}
660 \newcommand*\Frlabelitemiv{\FrenchLabelItem}
```

`\bbl@frenchlabelitems` `\bbl@frenchlabelitems` saves current itemize labels and changes them to their value in French. This code should never be executed twice in a row, so we need a new flag that will be set and reset by `\bbl@nonfrenchlabelitems` and `\bbl@frenchlabelitems`. `\bbl@frenchlabelitems` also sets reasonable default values for lengths `\labelwidthFB` and `\listindentFB` to be defined below, but these lengths can still be customised in the preamble.

```
661 \newif\iffB@enterFrench \FB@enterFrenchtrue
662 \def\bbl@frenchlabelitems{%
663   \iffB@enterFrench
664     \let\@ltiORI\labelitemi
665     \let\@ltiiORI\labelitemii
666     \let\@ltiiiORI\labelitemiii
667     \let\@ltivORI\labelitemiv
668     \let\labelitemi\Frlabelitemi
669     \let\labelitemii\Frlabelitemii
670     \let\labelitemiii\Frlabelitemiii
671     \let\labelitemiv\Frlabelitemiv
672     \ifdim\labelwidthFB<\z@
673       \settowidth{\labelwidthFB}{\FrenchLabelItem}%
674     \fi
675     \ifdim\listindentFB<\z@
676       \ifdim\parindent=\z@
677         \setlength{\listindentFB}{1.5em}%
678       \else
679         \setlength{\listindentFB}{\parindent}%
680       \fi
681     \fi
682     \FB@enterFrenchfalse
683   \fi
```

```

684 }
685 \def\bb1@nonfrenchlabelitems{%
686   \ifFB@enterFrench
687   \else
688     \let\Frlabelitemi\labelitemi
689     \let\Frlabelitemii\labelitemii
690     \let\Frlabelitemiii\labelitemiii
691     \let\Frlabelitemiv\labelitemiv
692     \let\labelitemi\@ltiORI
693     \let\labelitemii\@ltiiORI
694     \let\labelitemiii\@ltiiiORI
695     \let\labelitemiv\@ltivORI
696     \FB@enterFrenchtrue
697   \fi
698 }

```

`\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 `\bb1@frenchlabelitems`) unless they have been customised.

```

699 \newlength\listindentFB
700 \setlength{\listindentFB}{-1pt}
701 \newlength\labelwidthFB
702 \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.

```

703 \newlength\leftmarginFB
704 \def\FB@listHsettings{%
705   \leftmarginFB\labelwidthFB
706   \advance\leftmarginFB \labelsep
707   \leftmargini\leftmarginFB
708   \advance\leftmargini \listindentFB
709   \leftmarginii\leftmarginFB
710   \leftmarginiii\leftmarginFB
711   \leftmarginiv\leftmarginFB
712   \leftmargin\csname leftmargin\romannumeral\the\@listdepth\endcsname
713 }

```

`\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).

```

714 \def\FB@itemizesettings{%
715   \ifFBReduceListSpacing
716     \setlength{\itemsep}{\z@}%
717     \setlength{\parsep}{\z@}%
718     \setlength{\topsep}{\z@}%
719     \setlength{\partopsep}{\z@}%
720     \@tempdima=\parskip

```

```

721     \addtolength{\topsep}{-\@tempdima}%
722     \addtolength{\partopsep}{\@tempdima}%
723   \fi
724   \settowidth{\labelwidth}{\csname\@itemitem\endcsname}%
725   \ifFBListOldLayout
726     \setlength{\leftmargin}{\labelwidth}%
727     \addtolength{\leftmargin}{\labelsep}%
728     \addtolength{\leftmargin}{\parindent}%
729   \else
730     \FB@listHsettings
731   \fi
732 }

```

The definition of `\itemizeFB` follows the one of `\itemize` in standard L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> classes (see `ltlists.dtx`), spaces are customised by `\FB@itemizesettings`. `\enditemizeFB` will be defined as `\endlist=\endlistORI` ‘AtBeginDocument’.

```

733 \def\itemizeFB{%
734   \ifnum \@itemdepth >\thr@@\toodeep\else
735     \advance\@itemdepth\@ne
736     \edef\@itemitem{\labelitem\romannumeral\the\@itemdepth}%
737     \expandafter
738     \listORI
739     \csname\@itemitem\endcsname
740     \FB@itemizesettings
741   \fi}

```

`\enumerateFB` The definition of `\enumerateFB`, new to version 2.6a, follows the one of `\enumerate` in standard L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> classes (see `ltlists.dtx`), vertical spaces are customised (or not) via `\list` (`=\listFB` or `\listORI`) and horizontal spaces (leftmargins) are borrowed from `itemize` lists via `\FB@listHsettings`. `\endenumerateFB` will be defined as `\endlist=\endlistORI` ‘AtBeginDocument’.

```

742 \def\enumerateFB{%
743   \ifnum \@enumdepth >\thr@@\toodeep\else
744     \advance\@enumdepth\@ne
745     \edef\@enumctr{enum\romannumeral\the\@enumdepth}%
746     \expandafter
747     \list
748     \csname label\@enumctr\endcsname
749     {\FB@listHsettings
750      \usecounter\@enumctr\def\makelabel##1{\hss\llap{##1}}}%
751   \fi}

```

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

```

752 \def\descriptionFB{%
753   \list{}\{\FB@listHsettings
754     \labelwidth\z@ \itemindent-\leftmargin
755     \let\makelabel\descriptionlabel}}

```

## 2.9 French indentation of sections

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

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

```

756 \def\bbl@frenchindent{\let\@afterindentfalse\@afterindenttrue
757                      \@afterindenttrue}
758 \def\bbl@nonfrenchindent{\let\@afterindentfalse\@aifORI
759                      \@afterindentfalse}

```

## 2.10 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 2.13). Notice that 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` (whose definition 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.

```

760 \AtBeginDocument{\@ifpackageloaded{bigfoot}%
761                  {\FBWarning{frenchb.ldf}%
762                   {bigfoot package in use.\MessageBreak
763                    frenchb will NOT customise footnotes;\MessageBreak
764                     reported}}}%
765                  {\let\@footnotemarkORI\@footnotemark
766                   \def\@footnotemarkFB{\leavevmode\unskip\unkern
767                                         \,\@footnotemarkORI}%
768                   \ifFBAutoSpaceFootnotes
769                     \let\@footnotemark\@footnotemarkFB
770                     \fi}%
771                  }

```

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).

```

772 \newcommand*{\dotFFN}{.}
773 \newcommand*{\kernFFN}{\kern .5em}
774 \newdimen\parindentFFN
775 \parindentFFN=10in
776 \def\ftnISSymbol{\@fnsymbol\c@footnote}
777 \long\def\@makefntextFB#1{\ifx\thefootnote\ftnISSymbol

```



```

778             \@makefntextORI{#1}%
779         \else
780             \parindent=\parindentFFN
781             \rule\z@\footnotesep
782             \setbox\@tempboxa\hbox{\@thefnmark}%
783             \ifdim\wd\@tempboxa>\z@
784                 \llap{\@thefnmark}\dotFFN\kernFFN
785             \fi #1
786         \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).

```

787 \AtBeginDocument{\@ifpackageloaded{bigfoot}{}%
788     {\ifdim\parindentFFN<10in
789     \else
790         \parindentFFN=\parindent
791         \ifdim\parindentFFN<1.5em\parindentFFN=1.5em\fi
792     \fi
793     \let\@makefntextORI\@makefntext
794     \long\def\@makefntext#1{%
795         \ifFBFrenchFootnotes
796             \@makefntextFB{#1}%
797         \else
798             \@makefntextORI{#1}%
799         \fi}%
800     }%
801 }

```

For compatibility reasons, we provide definitions for the commands dealing with the layout of footnotes in `frenchb` version 1.6. `\frenchbsetup{}` (see in section 2.13) 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`.

```

802 \newcommand*{\AddThinSpaceBeforeFootnotes}{\FBAutoSpaceFootnotestruer}
803 \newcommand*{\FrenchFootnotes}{\FBFrenchFootnotestruer}
804 \newcommand*{\StandardFootnotes}{\FBFrenchFootnotesfalse}

```

## 2.11 Global layout

In multilingual documents, some typographic rules must depend on the current language (e.g., hyphenation, typesetting of numbers, spacing before double punctuation...), others should, IMHO, be kept global to the document: especially the layout of lists (see 2.8) and footnotes (see 2.10), and the indentation of the first paragraph of sections (see 2.9).

From version 2.2 on, if `frenchb` is `babel`’s “main language” (i.e. last language option at `babel`’s loading), `frenchb` customises the layout (i.e. lists, indentation of the first paragraphs of sections and footnotes) in the whole document regardless the current language. On the other hand, if `frenchb` is *not* `babel`’s “main language”, it leaves the layout unchanged both in French and in other languages.

`\FrenchLayout`    The former commands `\FrenchLayout` and `\StandardLayout` are kept for compatibility reasons but should no longer be used.

```

805 \newcommand*{\FrenchLayout}{%
806     \FBGlobalLayoutFrenchtrue
807     \PackageWarning{frenchb.ldf}%
808     {\protect\FrenchLayout\space is obsolete. Please use\MessageBreak
809     \protect\frenchbsetup{GlobalLayoutFrench} instead.}%
810 }
811 \newcommand*{\StandardLayout}{%
812     \FBReduceListSpacingfalse
813     \FBCompactItemizefalse
814     \FBStandardItemLabelstrue
815     \FBIndentFirstfalse
816     \FBFrenchFootnotesfalse
817     \FBAutoSpaceFootnotesfalse
818     \PackageWarning{frenchb.ldf}%
819     {\protect\StandardLayout\space is obsolete. Please use\MessageBreak
820     \protect\frenchbsetup{StandardLayout} instead.}%
821 }
822 \@onlypreamble\FrenchLayout
823 \@onlypreamble\StandardLayout

```

## 2.12 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- $\text{\LaTeX}$  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.

```

824 \ifFBunicode
825     \let\FBtextellipsis\textellipsis
826 \else
827     \DeclareTextSymbol{\FBtextellipsis}{LY1}{133}
828     \DeclareTextCommandDefault{\FBtextellipsis}{%
829         .\kern\fontdimen3\font.\kern\fontdimen3\font.\xspace}
830 \fi

```

`\Mdots@` and `\Tdots@ORI` hold the definitions of `\dots` in Math and Text mode. They default to those of `amsmath-2.0`, and will revert to standard  $\text{\LaTeX}$  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.

```

831 \newcommand*{\Tdots@ORI}{\@xp\textellipsis}
832 \newcommand*{\Tdots@}{\Tdots@ORI}
833 \newcommand*{\Mdots@}{\@xp\mdots@}
834 \AtBeginDocument{\DeclareRobustCommand*{\dots}{\relax
835     \csname\ifmmode M\else T\fi dots@\endcsname}%
836     \@ifundefined{@xp}{\let\@xp\relax}{}}%
837     \@ifundefined{mdots@}{\let\Tdots@ORI\textellipsis
838     \let\Mdots@\mathellipsis}

```

```

839 \let\mdots@\undefined\}}}
840 \def\bbl@frenchdots{\let\Tdots@\FBtextellipsis}
841 \def\bbl@nonfrenchdots{\let\Tdots@\Tdots@ORI}
842 \addto\extrasfrench{\bbl@frenchdots}
843 \addto\noextrasfrench{\bbl@nonfrenchdots}

```

## 2.13 Setup options: keyval stuff

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

```

844 \newif\ifFBStandardLayout \FBStandardLayouttrue
845 \newif\ifFBGlobalLayoutFrench \FBGlobalLayoutFrenchfalse
846 \newif\ifFBReduceListSpacing \FBReduceListSpacingfalse
847 \newif\ifFBListOldLayout \FBListOldLayouttrue
848 \newif\ifFBCompactItemize \FBCompactItemizefalse
849 \newif\ifFBStandardItemizeEnv \FBStandardItemizeEnvtrue
850 \newif\ifFBStandardEnumerateEnv \FBStandardEnumerateEnvtrue
851 \newif\ifFBStandardItemLabels \FBStandardItemLabelstrue
852 \newif\ifFBStandardLists \FBStandardListstrue
853 \newif\ifFBIndentFirst \FBIndentFirstfalse
854 \newif\ifFBFrenchFootnotes \FBFrenchFootnotesfalse
855 \newif\ifFBAutoSpaceFootnotes \FBAutoSpaceFootnotesfalse
856 \newif\ifFBOriginalTypewriter \FBOriginalTypewriterfalse
857 \newif\ifFBThinColonSpace \FBThinColonSpacefalse
858 \newif\ifFBThinSpaceInFrenchNumbers \FBThinSpaceInFrenchNumbersfalse
859 \newif\ifFBFrenchSuperscripts \FBFrenchSuperscriptstrue
860 \newif\ifFBLowercaseSuperscripts \FBLowercaseSuperscriptstrue
861 \newif\ifFBPartNameFull \FBPartNameFulltrue
862 \newif\ifFBSuppressWarning \FBSuppressWarningfalse
863 \newif\ifFBShowOptions \FBShowOptionsfalse

```

The defaults values of these flags have been set 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{}`.

```

864 \def\FB@french{french}
865 \AtEndOfPackage{%
866   \ifx\bbl@main@language\FB@french
867     \FBReduceListSpacingtrue
868     \FBListOldLayoutfalse
869     \FBCompactItemizetrue
870     \FBStandardItemizeEnvfalse
871     \FBStandardEnumerateEnvfalse
872     \FBStandardItemLabelsfalse
873     \FBIndentFirsttrue
874     \FBFrenchFootnotetrue
875     \FBAutoSpaceFootnotetrue
876     \FBGlobalLayoutFrenchtrue
877   \fi
878 }

```

`\frenchbsetup` From version 2.0 on, all setup options are handled by *one* command `\frenchbsetup` using the keyval syntax. Let's now define this command which reads and sets the options to be processed later (at `\begin{document}`) by `\FBprocess@options`. It can only be called in the preamble.

```
879 \newcommand*{\frenchbsetup}[1]{%
880   \setkeys{FB}{#1}%
881 }%
882 \@onlypreamble\frenchbsetup
```

`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.

`StandardLayout` resets the layout in French to the standard layout defined par the `LATEX` class and packages loaded. It deals with lists, indentation of first paragraphs of sections and footnotes. Other keys, entered *after* `StandardLayout` in `\frenchbsetup`, can overrule some of the `StandardLayout` settings.

When French is the main language, `GlobalLayoutFrench` forces the layout in French and (as far as possible) outside French to meet the French typographic standards.

```
883 \AtEndOfPackage{%
884   \RequirePackage{keyval}%
885   \define@key{FB}{StandardLayout}[true]%
886           {\csname FBStandardLayout#1\endcsname
887            \ifFBStandardLayout
888              \FBReduceListSpacingfalse
889              \FBStandardItemizeEnvtrue
890              \FBStandardEnumerateEnvtrue
891              \FBStandardItemLabelstrue
892              \FBIndentFirstfalse
893              \FBFrenchFootnotesfalse
894              \FBAutoSpaceFootnotesfalse
895              \FBGlobalLayoutFrenchfalse
896            \else
897              \FBReduceListSpacingtrue
898              \FBStandardItemizeEnvfalse
899              \FBStandardEnumerateEnvfalse
900              \FBStandardItemLabelsfalse
901              \FBIndentFirsttrue
902              \FBFrenchFootnotesttrue
903              \FBAutoSpaceFootnotesttrue
904            \fi}%
905   \define@key{FB}{GlobalLayoutFrench}[true]%
906           {\csname FBGlobalLayoutFrench#1\endcsname
907            \ifFBGlobalLayoutFrench
908              \ifx\bb1@main@language\FB@french
909                \FBReduceListSpacingtrue
910                \FBStandardItemizeEnvfalse
911                \FBStandardEnumerateEnvfalse
912                \FBStandardItemLabelsfalse
913                \FBIndentFirsttrue
914                \FBFrenchFootnotesttrue
915                \FBAutoSpaceFootnotesttrue
916              \else
```

```

917             \PackageWarning{frenchb.ldb}%
918             {Option 'GlobalLayoutFrench' skipped:
919             \MessageBreak French is *not*
920             babel's last option.\MessageBreak}%
921         \fi
922     \fi}%
923 \define@key{FB}{ReduceListSpacing}[true]%
924     {\csname FBReduceListSpacing#1\endcsname}%
925 \define@key{FB}{ListOldLayout}[true]%
926     {\csname FBListOldLayout#1\endcsname
927     \ifFBListOldLayout
928     \renewcommand*{\FrenchLabelItem}{\textendash}%
929     \fi}%
930 \define@key{FB}{CompactItemize}[true]%
931     {\csname FBCompactItemize#1\endcsname
932     \ifFBCompactItemize
933     \FBStandardItemizeEnvfalse
934     \FBStandardEnumerateEnvfalse
935     \else
936     \FBStandardItemizeEnvtrue
937     \FBStandardEnumerateEnvtrue
938     \fi}%
939 \define@key{FB}{StandardItemizeEnv}[true]%
940     {\csname FBStandardItemizeEnv#1\endcsname}%
941 \define@key{FB}{StandardEnumerateEnv}[true]%
942     {\csname FBStandardEnumerateEnv#1\endcsname}%
943 \define@key{FB}{StandardItemLabels}[true]%
944     {\csname FBStandardItemLabels#1\endcsname}%
945 \define@key{FB}{ItemLabels}{%
946     \renewcommand*{\FrenchLabelItem}{#1}}%
947 \define@key{FB}{ItemLabeli}{%
948     \renewcommand*{\Frlabelitemi}{#1}}%
949 \define@key{FB}{ItemLabelii}{%
950     \renewcommand*{\Frlabelitemii}{#1}}%
951 \define@key{FB}{ItemLabeliii}{%
952     \renewcommand*{\Frlabelitemiii}{#1}}%
953 \define@key{FB}{ItemLabeliv}{%
954     \renewcommand*{\Frlabelitemiv}{#1}}%
955 \define@key{FB}{StandardLists}[true]%
956     {\csname FBStandardLists#1\endcsname
957     \ifFBStandardLists
958     \FBReduceListSpacingfalse
959     \FBCompactItemizefalse
960     \FBStandardItemizeEnvtrue
961     \FBStandardEnumerateEnvtrue
962     \FBStandardItemLabelstrue
963     \else
964     \FBReduceListSpacingtrue
965     \FBCompactItemizetrue
966     \FBStandardItemizeEnvfalse
967     \FBStandardEnumerateEnvfalse
968     \FBStandardItemLabelsfalse
969     \fi}%
970 \define@key{FB}{IndentFirst}[true]%

```

```

971         {\csname FBIndentFirst#1\endcsname}%
972 \define@key{FB}{FrenchFootnotes}[true]%
973         {\csname FBFrenchFootnotes#1\endcsname}%
974 \define@key{FB}{AutoSpaceFootnotes}[true]%
975         {\csname FBAutoSpaceFootnotes#1\endcsname}%
976 \define@key{FB}{AutoSpacePunctuation}[true]%
977         {\csname FBAutoSpacePunctuation#1\endcsname}%
978 \define@key{FB}{OriginalTypewriter}[true]%
979         {\csname FBOriginalTypewriter#1\endcsname}%
980 \define@key{FB}{ThinColonSpace}[true]%
981         {\csname FBThinColonSpace#1\endcsname}%
982 \define@key{FB}{ThinSpaceInFrenchNumbers}[true]%
983         {\csname FBThinSpaceInFrenchNumbers#1\endcsname}%
984 \define@key{FB}{FrenchSuperscripts}[true]%
985         {\csname FBFrenchSuperscripts#1\endcsname}%
986 \define@key{FB}{LowercaseSuperscripts}[true]%
987         {\csname FBLowercaseSuperscripts#1\endcsname}%
988 \define@key{FB}{PartNameFull}[true]%
989         {\csname FBPartNameFull#1\endcsname}%
990 \define@key{FB}{SuppressWarning}[true]%
991         {\csname FBSuppressWarning#1\endcsname
992         \ifFBSuppressWarning
993         \renewcommand{\FBWarning}[2]{\relax}%
994         \else
995         \renewcommand{\FBWarning}[2]{%
996             \PackageWarning{##1}{##2}}%
997         \fi}
998 \define@key{FB}{ShowOptions}[true]%
999         {\csname FBShowOptions#1\endcsname}%

```

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`. The purpose of the following code is to map the French quote characters to `\og\ignorespaces` and `\fg` respectively when the current language is French, and to `\guillemotleft` and `\guillemotright` otherwise (think of German quotes); thus correct unbreakable spaces will be added automatically to French quotes. The quote characters typed in depend on the input encoding, it can be single-byte (latin1, latin9, applemac,...) or multi-bytes (utf-8, utf8x). We first check whether XeTeX or LuaTeX engines are used, if not the `inputenc` package has to be loaded before the `\begin{document}` with the proper coding option, so we check if `\DeclareInputText` is defined.

```

1000 \define@key{FB}{og}{%
1001     \newcommand*{\FB@og}{%
1002         \iflanguage{french}%
1003         {\ifFBAutoSpaceGuill\FB@og\ignorespaces
1004         \else\guillemotleft
1005         \fi}%
1006         {\guillemotleft}}%
1007 \ifFBunicode

```

LuaTeX or XeTeX in use,

```

1008 \ifFB@xetex@punct

```

`\XeTeXinterchartokenstate` is defined, we just need to set `\XeTeXcharclass` to `\FB@punctguillo` for the French opening quote (see subsection 2.2) and to switch

`\ifFBog@addspace` to false, otherwise commands `\og` and `\fg` would produce a double space; the `\ifFBguillo@addspace` flag is needed when switching from `\ttfamily` back to `\sf` or `\rm`.

```

1009         \XeTeXcharclass"00AB = \FB@punctguilo
1010         \FBguillo@addspacetrue \FBog@addspacefalse
1011     \else

```

then LuaTeX or an old XeTeX in use, the following trick for defining the active quote character is borrowed from `inputenc.dtx`.

```

1012         \catcode'#1=\active
1013         \bgroup
1014         \uccode'\~'#1%
1015         \uppercase{%
1016         \egroup
1017         \def~%
1018         }{\FB@@og}%
1019     \fi
1020 \else

```

This is for conventional TeX engines:

```

1021     \AtBeginDocument
1022     {\@ifundefined{DeclareInputText}%
1023      {\PackageWarning{frenchb.ldf}%
1024       {Option 'og' requires package inputenc.\MessageBreak}%
1025       \let\DeclareInputText\undefined
1026      }%
1027     {\@ifundefined{uc@dcclc}%

```

if `\uc@dcclc` is undefined, utf8x is not loaded...

```

1028         {\@ifundefined{DeclareUnicodeCharacter}%

```

if `\DeclareUnicodeCharacter` is undefined, utf8 is not loaded either, we assume 8-bit character input encoding. Package `MULEenc.sty` (from CJK) defines `\mule@def` to map characters to control sequences.

```

1029         {\@tempcnta'#1\relax
1030         \ifundefined{mule@def}%
1031         {\DeclareInputText{\the\@tempcnta}{\FB@@og}%
1032          \let\mule@def\undefined}%
1033         {\mule@def{11}{\FB@@og}}}%
1034         \let\DeclareUnicodeCharacter\undefined
1035     }%

```

utf8 loaded, use `\DeclareUnicodeCharacter`,

```

1036         {\DeclareUnicodeCharacter{00AB}{\FB@@og}}%
1037         \let\uc@dcclc\undefined
1038     }%

```

utf8x loaded, use `\uc@dcclc`,

```

1039         {\uc@dcclc{171}{default}{\FB@@og}}%
1040     }%
1041 }%
1042 \fi
1043 }%

```

Same code for the closing quote.

```

1044     \define@key{FB}{fg}{%

```

```

1045     \newcommand*{\FB@fg}{%
1046         \iflanguage{french}%
1047             {\ifFBAutoSpaceGuill\FB@fg
1048              \else\guillemotright
1049              \fi}%
1050             {\guillemotright}}%
1051     \ifFBunicode
1052     \ifFB@xetex@punct
1053         \XeTeXcharclass"00BB = \FB@punctguilf
1054         \FBguilf@addspace true \FBfg@addspace false
1055     \else
1056         \catcode'#1=\active
1057         \bgroup
1058         \uccode'~'#1%
1059         \uppercase{%
1060             \egroup
1061             \def~%
1062             }{{\FB@fg}}%
1063     \fi
1064 \else
1065     \AtBeginDocument
1066     {\@ifundefined{DeclareInputText}%
1067      {\PackageWarning{frenchb.ldb}%
1068       {Option 'fg' requires package inputenc.\MessageBreak}%
1069      \let\DeclareInputText\undefined
1070      }%
1071      {\@ifundefined{uc@dclc}%
1072       {\@ifundefined{DeclareUnicodeCharacter}%
1073        {\@tempcnta'#1\relax
1074         \ifundefined{mule@def}%
1075             {\DeclareInputText{\the\@tempcnta}{{\FB@fg}}%
1076              \let\mule@def\undefined
1077              }%
1078              {\mule@def{27}{{\FB@fg}}}%
1079              \let\DeclareUnicodeCharacter\undefined
1080              }%
1081              {\DeclareUnicodeCharacter{00BB}{{\FB@fg}}%
1082              }%
1083              \let\uc@dclc\undefined
1084              }%
1085              {\uc@dclc{187}{default}{{\FB@fg}}}%
1086              }%
1087              }%
1088      \fi
1089     }%
1090 }

```

`\FBprocess@options` `\FBprocess@options` processes the options, it is called *once* at `\begin{document}`.

```

1091 \newcommand*{\FBprocess@options}{%

```

Nothing has to be done here for `StandardLayout` and `StandardLists` (the involved flags have already been set in `\frenchbsetup{}` or before (at babel's `EndOfPackage`).

The next three options deal with the layout of lists in French.



`ReduceListSpacing` reduces the vertical spaces in lists in French (done by changing `\list` to `\listFB`). When `GlobalLayoutFrench` is true (the default), the same is done outside French except for languages that force a different setting.

```

1092 \ifFBReduceListSpacing
1093   \addto\extrasfrench{\let\list\listFB
1094                       \let\endlist\endlistFB}%
1095   \addto\noextrasfrench{\ifFBGlobalLayoutFrench
1096                         \let\list\listFB
1097                         \let\endlist\endlistFB
1098                         \else
1099                         \let\list\listORI
1100                         \let\endlist\endlistORI
1101                         \fi}%
1102 \else
1103   \addto\extrasfrench{\let\list\listORI
1104                       \let\endlist\endlistORI
1105                       }%
1106   \addto\noextrasfrench{\let\list\listORI
1107                       \let\endlist\endlistORI
1108                       }%
1109 \fi

```

When `ListOldLayout=true`, set `StandardEnumerateEnv=true` (before v. 2.6a `enumerate` was not redefined).

```

1110 \ifBListOldLayout
1111   \FBStandardEnumerateEnvtrue
1112 \fi

```

To ensure compatibility with the `enumitem` package, when `enumitem` is loaded, force flags `StandardItemizeEnv` and `StandardEnumerateEnv` to true.

```

1113 \@ifpackageloaded{enumitem}{%
1114   \ifFBStandardItemizeEnv
1115   \else
1116     \FBWarning{frenchb.ldf}%
1117     {Setting StandardItemizeEnv=true for\MessageBreak
1118      compatibility with enumitem package,\MessageBreak}%
1119     \FBStandardItemizeEnvtrue
1120   \fi
1121   \ifFBStandardEnumerateEnv
1122   \else
1123     \FBWarning{frenchb.ldf}%
1124     {Setting StandardEnumerateEnv=true for\MessageBreak
1125      compatibility with enumitem package,\MessageBreak}%
1126     \FBStandardEnumerateEnvtrue
1127   \fi}%
1128 {}%

```

To ensure compatibility with the `paralist` package, when `paralist` is loaded; force flags `StandardItemizeEnv` and `StandardEnumerateEnv` to true.

```

1129 \@ifpackageloaded{paralist}{%
1130   \ifFBStandardItemizeEnv
1131   \else
1132     \FBWarning{frenchb.ldf}%
1133     {Setting StandardItemizeEnv=true for\MessageBreak
1134      compatibility with paralist package,\MessageBreak}%

```

```

1135     \FBStandardItemizeEnvtrue
1136   \fi
1137   \ifFBStandardEnumerateEnv
1138   \else
1139     \FBWarning{frenchb.1df}%
1140     {Setting StandardEnumerateEnv=true for\MessageBreak
1141      compatibility with paralist package,\MessageBreak}%
1142     \FBStandardEnumerateEnvtrue
1143   \fi}%
1144 {}%

```

To ensure compatibility with the `enumerate` package, when `enumerate` is loaded, force flag `StandardEnumerateEnv` to true.

```

1145 \@ifpackageloaded{enumerate}{%
1146   \ifFBStandardEnumerateEnv
1147   \else
1148     \FBWarning{frenchb.1df}%
1149     {Setting StandardEnumerateEnv=true for\MessageBreak
1150      compatibility with enumerate package,\MessageBreak}%
1151     \FBStandardEnumerateEnvtrue
1152   \fi}%
1153 {}%

```

Let's now process flags `StandardItemizeEnv` and `StandardEnumerateEnv`. Mind that `StandardEnumerateEnv`'s value decides for both environments `enumerate` and `description`).

```

1154 \ifFBStandardItemizeEnv
1155   \addto\extrasfrench{\let\itemize\itemizeORI}%
1156   \addto\noextrasfrench{\let\itemize\itemizeORI}%
1157 \else
1158   \addto\extrasfrench{\let\itemize\itemizeFB}%
1159   \addto\noextrasfrench{\ifFBGlobalLayoutFrench
1160     \let\itemize\itemizeFB
1161   \else
1162     \let\itemize\itemizeORI
1163   \fi}%
1164 \fi
1165 \ifFBStandardEnumerateEnv
1166   \addto\extrasfrench{\let\enumerate\enumerateORI
1167     \let\description\descriptionORI}%
1168   \addto\noextrasfrench{\let\enumerate\enumerateORI
1169     \let\description\descriptionORI}%
1170 \else
1171   \addto\extrasfrench{\let\enumerate\enumerateFB
1172     \let\description\descriptionFB}%
1173   \addto\noextrasfrench{\ifFBGlobalLayoutFrench
1174     \let\enumerate\enumerateFB
1175     \let\description\descriptionFB
1176   \else
1177     \let\enumerate\enumerateORI
1178     \let\description\descriptionORI
1179   \fi}%
1180 \fi

```

`StandardItemLabels` resets `labelitems` in French to their standard values set by

the L<sup>A</sup>T<sub>E</sub>X class and packages loaded. When `GlobalLayoutFrench` is true, labelitems are identical whether the current language is French or not.

```

1181 \iffBStandardItemLabels
1182 \addto\extrasfrench{\bbl@nonfrenchlabelitems}%
1183 \addto\noextrasfrench{\bbl@nonfrenchlabelitems}%
1184 \else
1185 \addto\extrasfrench{\bbl@frenchlabelitems}%
1186 \addto\noextrasfrench{\iffBGlobalLayoutFrench
1187 \bbl@frenchlabelitems
1188 \else
1189 \bbl@nonfrenchlabelitems
1190 \fi}%
1191 \fi

```

`IndentFirst` forces the first paragraphs of sections to be indented just like the other ones in French. When `GlobalLayoutFrench` is true, the same is done outside French except for languages that force a different setting. `\bbl@nonfrenchindent` has been designed to be smart with other languages (like Spanish) who also indent every first paragraphs of sections (see section 2.9).

```

1192 \iffBIndentFirst
1193 \addto\extrasfrench{\bbl@frenchindent}%
1194 \addto\noextrasfrench{\iffBGlobalLayoutFrench
1195 \bbl@frenchindent
1196 \else
1197 \bbl@nonfrenchindent
1198 \fi}%
1199 \else
1200 \addto\extrasfrench{\bbl@nonfrenchindent}%
1201 \addto\noextrasfrench{\bbl@nonfrenchindent}%
1202 \fi

```

The layout of footnotes is handled at the `\begin{document}` depending on the values of flags `FrenchFootnotes` and `AutoSpaceFootnotes` (see section 2.10), 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.

```

1203 \iffBAutoSpacePunctuation
1204 \autospace@beforeFDP
1205 \else
1206 \noautospace@beforeFDP
1207 \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.

```

1208 \iffBOriginalTypewriter
1209 \else
1210 \let\ttfamily\ttfamilyFB
1211 \let\rmfamily\rmfamilyFB
1212 \let\sffamily\sffamilyFB
1213 \fi

```

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

```

1214 \ifFBThinColonSpace\renewcommand*{\FBcolonspace}{\FBthinspace}\fi
    When true, ThinSpaceInFrenchNumbers redefines numprint.sty's command
    \npstylefrench to set \npthousandsep to \, (thinspace) instead of ~ (de-
    fault) . This option has no effect if package numprint.sty is not loaded with
    'autolanguage'. As old versions of numprint.sty did not define \npstylefrench,
    we have to provide this command.
1215 \@ifpackageloaded{numprint}%
1216 {\ifnprt@autolanguage
1217   \providecommand*\npstylefrench{}}%
1218   \ifFBThinSpaceInFrenchNumbers
1219     \renewcommand*\npstylefrench{%
1220       \npthousandsep{\,}%
1221       \npdecimalsign{,}%
1222       \npproductsign{\cdot}%
1223       \npunitseparator{\,}%
1224       \npdegreeseperator{}}%
1225       \nppercentseparator{\nprt@unitsep}%
1226     }%
1227   \else
1228     \renewcommand*\npstylefrench{%
1229       \npthousandsep{~}%
1230       \npdecimalsign{,}%
1231       \npproductsign{\cdot}%
1232       \npunitseparator{\,}%
1233       \npdegreeseperator{}}%
1234       \nppercentseparator{\nprt@unitsep}%
1235     }%
1236   \fi
1237   \npaddtolanguage{french}{french}%
1238 \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.

```

1239 \ifFBFrenchSuperscripts
1240   \DeclareRobustCommand*\up*{\@ifstar{\FB@up@fake}{\fup}}%
1241 \else
1242   \DeclareRobustCommand*\up*{\@ifstar{\FB@up@fake}%
1243                                   {\textsuperscript}}%
1244 \fi

```

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

```

1245 \ifFBLowercaseSuperscripts
1246 \else
1247   \renewcommand*\FB@lc[1]{##1}%
1248 \fi

```

PartNameFull: if false, redefine \partname.

```

1249 \ifFBPartNameFull
1250 \else\addto\captionsofrench{\def\partname{Partie}}\fi

```

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

```

1251 \ifFBSHOWOptions
1252   \GenericWarning{* }{%
1253     * **** List of possible options for frenchb ****\MessageBreak
1254     [Default values between brackets when frenchb is loaded *LAST*]%
1255     \MessageBreak
1256     ShowOptions=true [false]\MessageBreak
1257     StandardLayout=true [false]\MessageBreak
1258     GlobalLayoutFrench=false [true]\MessageBreak
1259     StandardLists=true [false]\MessageBreak
1260     IndentFirst=false [true]\MessageBreak
1261     ReduceListSpacing=false [true]\MessageBreak
1262     ListOldLayout=true [false]\MessageBreak
1263     StandardItemizeEnv=true [false]\MessageBreak
1264     StandardEnumerateEnv=true [false]\MessageBreak
1265     StandardItemLabels=true [false]\MessageBreak
1266     ItemLabels=\textendash, \textbullet,
1267       \protect\ding{43},... [\textendash]\MessageBreak
1268     ItemLabeli=\textendash, \textbullet,
1269       \protect\ding{43},... [\textendash]\MessageBreak
1270     ItemLabelii=\textendash, \textbullet,
1271       \protect\ding{43},... [\textendash]\MessageBreak
1272     ItemLabeliii=\textendash, \textbullet,
1273       \protect\ding{43},... [\textendash]\MessageBreak
1274     ItemLabeliv=\textendash, \textbullet,
1275       \protect\ding{43},... [\textendash]\MessageBreak
1276     FrenchFootnotes=false [true]\MessageBreak
1277     AutoSpaceFootnotes=false [true]\MessageBreak
1278     AutoSpacePunctuation=false [true]\MessageBreak
1279     OriginalTypewriter=true [false]\MessageBreak
1280     ThinColonSpace=true [false]\MessageBreak
1281     ThinSpaceInFrenchNumbers=true [false]\MessageBreak
1282     FrenchSuperscripts=false [true]\MessageBreak
1283     LowercaseSuperscripts=false [true]\MessageBreak
1284     PartNameFull=false [true]\MessageBreak
1285     SuppressWarning=true [false]\MessageBreak
1286     og= <left quote character>, fg= <right quote character>
1287     \MessageBreak
1288     *****
1289     \MessageBreak\protect\frenchbsetup{ShowOptions}}
1290   \fi
1291 }

```

At `\begin{document}` we save again the value of `\if@afterindent`, definitions of the list environments and the values of `labelitems` so that all changes made in the preamble are taken into account in languages other than French and in French with the `StandardLayout` option. We also have to provide an `\xspace` command in case the `xspace.sty` package is not loaded.

```

1292 \AtBeginDocument{%
1293   \ifx\@afterindentfalse\@afterindenttrue
1294     \let\@aifORI\@afterindenttrue
1295   \else \let\@aifORI\@afterindentfalse
1296   \fi
1297   \let\listORI\list
1298   \let\endlistORI\endlist

```

```

1299 \let\endlistFB\endlist
1300 \let\itemizeORI\itemize
1301 \let\enditemizeORI\endlist
1302 \let\enditemizeFB\endlist
1303 \let\enumerateORI\enumerate
1304 \let\endenumerateORI\endlist
1305 \let\endenumerateFB\endlist
1306 \let\descriptionORI\description
1307 \let\enddescriptionORI\endlist
1308 \let\enddescriptionFB\endlist
1309 \let\@ltiORI\labelitemi
1310 \let\@ltiiORI\labelitemii
1311 \let\@ltiiiORI\labelitemiii
1312 \let\@ltivORI\labelitemiv
1313 \providecommand*{\xspace}{\relax}%

```

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

```

1314 \@ifundefined{pdfstringdefDisableCommands}{%
1315   {\let\pdfstringdefDisableCommands\undefined}%
1316   {\pdfstringdefDisableCommands{%
1317     \let\up\relax
1318     \let\up\relax
1319     \let\degre\textdegree
1320     \let\degres\textdegree
1321     \def\ieme{e\xspace}%
1322     \def\iemes{es\xspace}%
1323     \def\ier{er\xspace}%
1324     \def\iers{ers\xspace}%
1325     \def\iere{re\xspace}%
1326     \def\ieres{res\xspace}%
1327     \def\FrenchEnumerate#1{#1\degre\space}%
1328     \def\FrenchPopularEnumerate#1{#1\degre)\space}%
1329     \def\No{N\degre\space}%
1330     \def\no{n\degre\space}%
1331     \def\Nos{N\degre\space}%
1332     \def\nos{n\degre\space}%
1333     \def\FB@og{\guillemotleft\space}%
1334     \def\FB@fg{\space\guillemotright}%
1335     \def\at{@}%
1336     \def\circonflexe{\string~}%
1337     \def\tild{\string~}%
1338     \let\bsc\textsc
1339     }%
1340   }%

```

It is time to process the options set with `\frenchboptions{}`. Then we need to execute either `\extrasfrench` and `\captionsfrench` or `\noextrasfrench` according to the current language at the `\begin{document}` (these three commands have been updated by `\FBprocess@options`). *But*, when French is the main language, `\extrasfrench` is executed *again* now (French has been switched on 'AtBeginDocument' some time before). This is harmless, except for `\bbl@frenchspacing` which will redefine `\bbl@nonfrenchspacing` to `\relax`, this will be wrong if the user switches to English. When French is *not* the main language, `\noextrasfrench` executes `\bbl@nonfrenchspacing`

(`=\nonfrenchspacing`), thus eventually overwriting a `\frenchspacing` command issued by the main language (German, Spanish, etc.). So we have to define `\bbl@nonfrenchspacing` as `\relax` here and restore it's meaning afterwards. It is also the right time, when a koma-script class is in use, to save the value of `\captionformat` in `\CaptionSeparatorORI` and to add the redefinitions of `\captionformat` to `\(no)extrasfrench`.

```

1341 \FBprocess@options
1342 \let\bbl@nonfrenchspacingORI\bbl@nonfrenchspacing
1343 \let\bbl@nonfrenchspacing\relax
1344 \@ifundefined{captionformat}%
1345   {\let\captionformat\undefined}%
1346   {\let\CaptionSeparatorORI\captionformat
1347     \addto\extrasfrench{%
1348       \renewcommand*{\captionformat}{\CaptionSeparator}}%
1349     \addto\noextrasfrench{%
1350       \renewcommand*{\captionformat}{\CaptionSeparatorORI}}%
1351   }%
1352 \ifx\bbl@main@language\FB@french
1353   \extrasfrench\captionsfrench
1354 \else
1355   \noextrasfrench
1356 \fi
1357 \let\bbl@nonfrenchspacing\bbl@nonfrenchspacingORI

```

Some warnings are issued when output font encodings are not properly set. With XeLaTeX, `fontspec.sty` and `xunicode.sty` should be loaded; 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!

```

1358 \ifFBXeTeX
1359   \@ifundefined{DeclareUTFcharacter}%
1360     {\PackageWarning{frenchb.ldf}%
1361       {Add \protect\usepackage{xltextra} to the\MessageBreak
1362         preamble of your document,}%
1363     \let\DeclareUTFcharacter\undefined}%
1364   {}%
1365 \else
1366   \begingroup \newcommand{\FBOTone}{OT1}%
1367   \ifx\encodingdefault\FBOTone
1368     \PackageWarning{frenchb.ldf}%
1369       {OT1 encoding should not be used for French.
1370       \MessageBreak
1371       Add \protect\usepackage[T1]{fontenc} to the
1372       preamble\MessageBreak of your document,}%
1373   \fi
1374 \endgroup
1375 \fi
1376 }

```

## 2.14 Clean up and exit

Load `frenchb.cfg` (kept just for backward compatibility).

```

1377 \loadlocalcfg{frenchb}

```

Final cleaning. The macro `\ldf@quit` takes care for setting the main language to be switched on at `\begin{document}` and resetting the category code of `@` to its original value. The config file searched for has to be `frenchb.cfg`, and `\CurrentOption` has been set to 'french', so `\ldf@finish\CurrentOption` cannot be used: we first load `frenchb.cfg`, then call `\ldf@quit\CurrentOption`.

```
1378 \FBclean@on@exit
1379 \ldf@quit\CurrentOption
```



## Change History

v2.0	General: <code>\parindentFFN</code> not changed if already defined (required by JA for cah-gut.cls). . . . . 32	v2.0d	General: Options <code>og</code> and <code>fg</code> changed: limit the definition to French so that quote characters can be used in German. . . . . 35
	Added warning for OT1 encoding. . . . . 45	v2.0e	General: New option: <code>StandardLists</code> . . . . . 35
	<code>AtBeginDocument</code> , save again the definitions of the ‘list’ and ‘itemize’ environments and the values of <code>labelitems</code> . As of frenchb v. 1.6, ‘ORI’ values were set when reading frenchb.ldf, later changes were ignored. . . 45	v2.0f	General: <code>StandardLayout</code> option had no effect on lists. Test moved to <code>\FBprocess@options</code> . . . . . 35
	Footnotes are now printed by default ‘à la française’ for the whole document. . . . . 32		Two typos corrected in option <code>StandardLists</code> : <code>[false] → [true]</code> and <code>StandardLayout → StandardLists</code> . . . . . 35
	New command <code>\frenchbsetup</code> added for global customisation. 35	v2.0g	General: 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> . . . . . 35
	<code>\bsc</code> : <code>\hbox</code> dropped, replaced by <code>\kern0pt</code> . . . . . 22		<code>\StandardLayout</code> : Flag <code>\ifFBStandardLayout</code> not checked by <code>\FBprocess@options</code> , low-level flags have to be set one by one. . . . . 33
	<code>\captionsfrench</code> : ‘Fig.’ changed to ‘Figure’ and ‘Tab.’ to ‘Table’. . . . . 25	v2.1a	General: Command <code>\fup</code> added to produce better superscripts than <code>\textsuperscript</code> . . . . 19
	Set <code>\CaptionSeparator</code> in <code>\extrasfrench</code> now instead of <code>\captionsfrench</code> because it has to be reset when leaving French. . . . . 25		New option: <code>FrenchSuperscripts</code> to define <code>\up</code> as <code>\fup</code> or as <code>\textsuperscript</code> . . . . . 35
	<code>\datefrench</code> : 2 ‘ <code>\relax</code> ’ added in <code>\today</code> ’s definition. . . . . 19		New option: <code>LowercaseSuperscripts</code> . . . . . 35
	<code>\FBtextellipsis</code> : Added special case for LY1 encoding, see bug report from Bruno Voisin (2004/05/18). . . . . 34		<code>\datefrench</code> : <code>\today</code> changed (correction in 2.0 was wrong: <code>\today</code> was printed without spaces in toc). . . . . 19
	<code>\nombre</code> : <code>\nombre</code> requires now <code>numprint.sty</code> . . . . . 24	v2.1b	General: Disable some commands in bookmarks. . . . . 45
v2.0b	General: Footnotes: Just do nothing (except warning) when the <code>bigfoot</code> package is loaded. . . . 32		
v2.0c	General: <code>\ThinSpaceInFrenchNumbers</code> added for compatibility with frenchb-1.x. . . . . 25		
	Option <code>ThinSpaceInFrenchNumbers</code> added. . . . . 35		
	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> . . . . . 25		

	\fup: Command \fup changed to use real superscripts from fourier v. 1.6. ....	19		and \AutoSpaceBeforeFDP now set the flag \iffBAutoSpacePunctuation accordingly (LaTeX only). ...	16
v2.1c	General: Added commands \Nos and \nos. ....	21		In LaTeX, frenchb no longer adds spaces before ‘double punctuation’ characters in computer code. Suggested by Yannis Haralambous. ....	17
	\degres: Provide a temporary definition (hyperref safe) of \degres in case it has to be expanded in the preamble (by beamer’s \title command for instance). ....	23		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. ....	35
	\up: Provide a temporary definition (hyperref safe) of \up in case it has to be expanded in the preamble (by beamer’s \title command for instance). ....	19		\fup: \lowercase changed to \MakeLowercase as the former doesn’t work for non ASCII characters in encodings like apple mac, utf-8,...	19
v2.1d	General: Argument of \ProvidesLanguage changed above from ‘french’ to ‘frenchb’ (otherwise \listfiles prints no date/version information). The real name of current language (french) as to be corrected before calling \LdfInit. ....	9	v2.3b	General: New commands \dotFFN and \kernFFN for more flexibility (suggested by JA). ....	32
	Avoid warning “\end occurred when \ifx ... incomplete” with LaTeX-2.09. ....	9	v2.3c	General: Commands \ttfamily, \rmfamily and \sffamily have to be robust. Bug introduced in 2.3a, pointed out by Manuel Pégourie-Gonnard. ....	17
v2.2a	General: 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. ....	35	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. ...	31
	Values of flags \ifBReduceListSpacing, \ifBCompactItemize, \ifBStandardItemLabels, \ifBIndentFirst, \ifBFrenchFootnotes, \ifBFAutoSpaceFootnotes changed: default now means ‘StandardLayout’, it will be changed to ‘FrenchLayout’ AtEndOfPackage only if french is \bbl@main@language. ....	35		\frenchbsetup: Warning added to \GlobalLayoutFrench when French is not the main language. ....	36
	When frenchb is babel’s last option, French becomes the document’s main language, so GlobalLayoutFrench applies. ....	35	v2.3e	General: Execute \AutoSpaceBeforeFDP also in LaTeX to define \FDP@colonspace: needed for tex4ht, pointed out by MPG. .	16
	\fup: \newif and \newdimen moved before \ifLaTeXe to avoid an error with plainTeX. ....	19	v2.4a	General: \PackageWarning changed to \FBWarning (in case \@makecaption has been customised). \FBWarning is defined as \PackageWarning by	
v2.3a	General: \NoAutoSpaceBeforeFDP				

	default but can be made silent using <code>\frenchbsetup</code> , (suggested by MPG). . . . .	26		New definitions needed for XeTeX/LuaTeX to properly print some dates and captions: using c.s. like <code>\’e</code> do not work with XeTeX (OK with XeLaTeX). . .	22
	<code>\PackageWarning</code> changed to <code>\FBWarning</code> (when bigfoot package in use). . . . .	32		Punctuation is no longer made active with XeTeX-based engines.	11
	New option <code>SuppressWarning</code> .	35		Test <code>\@ifundefined</code> leaves the tested control sequence defined as <code>\relax</code> when TRUE. Changed <code>\relax</code> to <code>\undefined</code> when testing <code>\numprint</code> . . . .	24
	<code>\ifFBXeTeX</code> : Added a new ‘if’ <code>\FBunicode</code> and some <code>\lccode</code> definitions to <code>\extrasfrench</code> and <code>\noextrasfrench</code> . . . . .	11		Test <code>\@ifundefined</code> leaves the tested control sequence defined as <code>\relax</code> when TRUE. Changed <code>\relax</code> to <code>\undefined</code> when testing <code>\pdfstringdefDisableCommands</code> at <code>AtBeginDocument</code> . . . . .	45
v2.4b	<code>\FBprocess@options</code> : Warning and set flag <code>CompactItemize</code> false when the <code>enumitem</code> package is loaded (ensures compatibility).	41		<code>\captionsfrench</code> : <code>\emph</code> deleted in <code>\seename</code> and <code>\alsoname</code> to match what is done for the other languages. Suggested by Marc Baudoin. . . . .	25
v2.4c	General: In <code>\ttfamilyFB</code> , also cancel automatic spaces inside French guillemets entered as characters (see <code>\frenchbsetup</code> ). . . . .	17		Replaced <code>\’e</code> , <code>\’e</code> and <code>\’a</code> by c.s. to work with XeTeX. . . . .	25
	<code>\frenchbsetup</code> : In <code>\ttfamilyFB</code> , also cancel automatic spaces inside French guillemets coded as characters (see <code>\frenchbsetup</code> ). . . . .	38		<code>\datefrench</code> : Replaced <code>\’e</code> and <code>\u</code> by c.s. to work with XeTeX and LuaTeX. . . . .	19
v2.4d	<code>\up</code> : Command <code>\up</code> defined with <code>\providecommand</code> instead of <code>\newcommand</code> as <code>\up</code> may be defined elsewhere ( <code>catalan.ldf</code> ). Bug pointed out by Felip Manyé i Ballester. . . . .	19		<code>\frenchbsetup</code> : Test <code>\@ifundefined</code> leaves the tested control sequence defined as <code>\relax</code> when TRUE. Changed <code>\relax</code> to <code>\undefined</code> when testing <code>\XeTeXrevision</code> , <code>\DeclareInputText</code> , <code>\uc@dcl</code> , <code>\DeclareUnicodeCharacter</code> , <code>\mule@def</code> in <code>\og</code> and <code>\fg</code> . . .	38
v2.5a	General: <code>\og</code> and <code>\fg</code> do not print correctly in English when using XeTeX or LuaTeX, fixed by using <code>\textquotedblleft</code> and <code>\textquotedblright</code> defined above. . . . .	19		<code>\textquoteddblright</code> : Change <code>\guillemotleft</code> and <code>\guillemotright</code> definitions for Unicode and provide definitions for <code>\textquotedblleft</code> and <code>\textquotedblright</code> . Insures correct printing of quotes by <code>\og</code> and <code>\fg</code> in French and outside. . . . .	18
	Define <code>\bbl@nonfrenchspacing</code> locally as <code>\relax</code> , otherwise the <code>\bbl@frenchspacing</code> command included in <code>germanb.ldf</code> is overwritten here by <code>\noextrasfrench</code> . Bug pointed out by Ulrike Fischer. . . . .	47		<code>\up</code> : Test <code>\@ifundefined</code> leaves the tested control sequence defined as <code>\relax</code> when TRUE. Changed <code>\relax</code> to <code>\undefined</code> when testing <code>\realsuperscript</code> . . . . .	19
	Define <code>\Fthinspace</code> for those who want to customise the width of the space before ; and ? . . . . .	12			
	New command <code>\NoAutoSpacing</code> , suggested by MPG. . . . .	17			

v2.5b	General: <code>\captionsfrench</code> will be executed ‘AtBeginDocument’ after <code>\FBprocess@options</code> , no need to add it here. . . . . 26	<code>\at</code> , <code>\circonflexe</code> , <code>\tild</code> , <code>\boi</code> and <code>\degre</code> for Unicode based engines. . . . . 22
	Do not use the test <code>\iflanguage{french}</code> to check whether French is the main language or not, as it might be erroneously positive when English is the main language and no hyphenation patterns are available for French. In this case <code>\l@french</code> and <code>\l@english</code> are 0. Pointed out by Günter Milde. . . . . 35, 47	<code>\FBtextellipsis</code> : Unicode fonts also provide a ready made character for <code>\textellipsis</code> , let’s just use it (reported by Maxime Chupin, 2011/06/04). . . . . 34
	<code>\frenchbsetup</code> : Do not use the test <code>\iflanguage{french}</code> to check whether French is the main language or not, as it might be erroneously positive when English is the main language and no hyphenation patterns are available for French. In this case <code>\l@french</code> and <code>\l@english</code> are 0. Pointed out by Günter Milde. . . . . 36	v2.5g
		General: <code>\FBguill@spacing</code> needs to be skipped in commands <code>\FB@og</code> and <code>\FB@fg</code> only when XeTeX’s “inter char” mechanism is triggered for quotes, see <code>\frenchbsetup</code> . . . . . 18
v2.5c	<code>\frenchbsetup</code> : The code meant for XeTeX also works for LuaTeX, we just need to change the test. 38	Add four <code>\newif</code> to control spacing of quotes (characters and control sequences). . . . . 12
		Redefine <code>\degre</code> , <code>\degres</code> <code>\at</code> <code>\circonflexe</code> and <code>\tild</code> for bookmarks. Add <code>\fup</code> also. . . 46
v2.5d	General: Moved the <code>\newcount</code> command outside <code>\ifFB@xetex@punct</code> ... <code>\fi</code> (it broke Plain formats). 12	Skip the message in Plain, <code>\PackageInfo</code> undefined for Plain formats. . . . . 12
	<code>\ifFBXeTeX</code> : Added two new ‘if’ <code>\FBXeTeX</code> and <code>\FBLuaTeX</code> as XeTeX and behave differently regarding the status of the French “apostrophe”. . . . . 11	Switch flags <code>\iffBog@addspace</code> and <code>\iffBfg@addspace</code> to true, otherwise <code>\og</code> and <code>\fg</code> provide no spacing in XeLaTeX with options ‘og’ and ‘fg’ activated in <code>\frenchbsetup</code> (bug v. 2.5a-f). 17
		<code>\FB@xetex@punct@french</code> : XeTeX-charclass(es) for French quotes will be set to <code>\FB@punctguillo</code> and <code>\FB@punctguilf</code> by options ‘og’ and ‘fg’ in <code>\frenchbsetup</code> . French quotes should behave as normal characters by default in XeLaTeX as in LaTeX. . . . . 13
v2.5e	General: <code>\pdfstringdefDisableCommands</code> should redefine <code>\FB@og</code> and <code>\FB@fg</code> instead of <code>\og</code> and <code>\fg</code> so that it works also when quotes are entered as characters. Reported by Sébastien Gouezel. . . . . 45	<code>\frenchbsetup</code> : When <code>\iffB@xetex@punct</code> is true, ‘og’ and ‘fg’ options now set XeTeXcharclasses of these characters to <code>\FB@punctguillo</code> and <code>\FB@punctguilf</code> . Otherwise French quotes behave as normal characters (their XeTeXcharclass is 0). . . . . 38
	The redefinition of <code>\pdfstringdefDisableCommands</code> from <code>\relax</code> to <code>\undefined</code> was misplaced. Reported by Sébastien Gouezel. . . . . 45	<code>\NoAutoSpacing</code> : Switch flags <code>\iffBog@addspace</code> and <code>\iffBfg@addspace</code> to true, otherwise <code>\og</code> and <code>\fg</code> provide no spacing in XeLaTeX with options ‘og’ and ‘fg’ activated in <code>\frenchbsetup</code> (bug v. 2.5a-f). 17
v2.5f	General: Changed definitions of	



must be reset, f.i. when an itemize list occurs inside en- vironments based on trivlist which set <code>\labelwidth</code> to 0 (see proof environment in am- sthm.sty). Bug pointed out by Julien Hauseux. . . . .	30	vent frenchb from adding it's own space before 'double punc- tuation' characters. . . . .	13
v2.6g		<code>\FB@itemizesettings</code> : Suppress all vertical spaces only if <code>ReduceListSpacing</code> is true. Pointed out by Pierre Willaime. . . . .	30
General: U+00A0 (Unicode no- breakspace) and U+202F (Uni- code nobreakthinspace) added to class <code>\FB@punctnul</code> to pre-		<code>\ifFBXeTeX</code> : lccode values for the French "apostrophe" are now the same for XeTeX and Lua- TeX. . . . .	11