

# Mode d'emploi du module `babel-french`

Daniel FLIPO

Version 4.0e – 15 août 2025

**Cette documentation concerne  
uniquement Lua(La)TeX.**

**Les utilisateurs de pdf(La)TeX et  
Xe(La)TeX sont invités à consulter  
le fichier `frenchb3-doc.pdf`.**

## Résumé

La première version de `frenchb` (**french** pour **babel**) est sortie en 1996. La version 2, profondément remaniée, date de mai 2007. Une version 3 a vu le jour en 2014 alors que LuaTeX n'était pas encore en version 1.0.

La version actuelle de `frenchb` (4.0e), dont le nom officiel est `babel-french`, a été scindée en deux parties distinctes : une version « historique » `frenchb3.dtx`, réservée aux moteurs anciens (TeX, pdfTeX et XeTeX) qui restera figée — sauf corrections éventuelles de bogues –, l'autre « moderne » `frenchb.dtx` destinée à Lua(La)TeX, qui continuera à évoluer.

**La liste détaillée des changements introduits dans la version 4.0 se trouve à la section 4.1 p. 16.**

## Historique des mises à jour de cette documentation

**28 juin 2025 :**      Adaptation à la version 4.0a, voir section 4.1.

# Sommaire

<b>1</b>	<b>Appel de l'extension Babel</b>	<b>3</b>
<b>2</b>	<b>Description de la francisation par babel-french</b>	<b>4</b>
<b>3</b>	<b>Personnalisation</b>	<b>7</b>
3.1	<code>\frenchsetup</code> . . . . .	8
3.2	Traduction des intitulés . . . . .	12
3.3	Présentation des listes . . . . .	13
<b>4</b>	<b>Changements entre les versions 4.0e et 3.6</b>	<b>16</b>
4.1	Quoi de neuf en version 4.0? . . . . .	16
4.2	Quoi de neuf en version 3.7? . . . . .	18
4.3	Quoi de neuf en version 3.6? . . . . .	18
<b>5</b>	<b>Problèmes de césures</b>	<b>18</b>
<b>6</b>	<b>Incompatibilités connues et remèdes</b>	<b>18</b>
<b>7</b>	<b>Bibliographie</b>	<b>19</b>
7.1	Bibliographie avec BibTeX . . . . .	19
7.2	Bibliographie avec Biblatex/Biber . . . . .	20
<b>8</b>	<b>Compatibilité avec e-french</b>	<b>20</b>

# 1 Appel de l'extension Babel

Babel est installé en standard dans toutes distributions LaTeX, pour disposer des langues française et anglaise<sup>1</sup>, il suffit d'ajouter `\usepackage[english, french]{babel}`<sup>2</sup> dans le préambule du document (entre `\documentclass` et `\begin{document}`).

Il est recommandé de déclarer les options de langues comme arguments de `\documentclass`, elles peuvent alors être utilisées également par d'autres extensions :

```
\documentclass[12pt, british, french]{article}
\usepackage{varioref}
\usepackage{babel}
```

a le même effet que

```
\documentclass[12pt]{article}
\usepackage[french]{varioref}
\usepackage[british, french]{babel}
```

La *dernière* langue chargée en option de Babel ou de `\documentclass` (le français dans les exemples ci-dessus) est la *langue principale* du document, c'est elle qui est active au début du document et qui régit la présentation générale (listes, notes de bas de page, retrait des premiers paragraphes) quelle que soit la langue courante.

Pour changer de langue en cours de document on utilise la commande standard de Babel `\selectlanguage{lang}`, par exemple `\selectlanguage{british}` et pour revenir en français `\selectlanguage{french}`<sup>3</sup>.

Pour passer *localement* dans une autre langue on peut utiliser l'environnement

```
\begin{otherlanguage}{langue}
  texte...
\end{otherlanguage}
```

ou pour une courte citation dans un paragraphe

```
\foreignlanguage{langue}{texte...}.
```

Une syntaxe allégée est proposée pour les changements de langue : en ajoutant par exemple dans le préambule `\babeltags{fr = french}`<sup>4</sup>, on peut remplacer

```
\foreignlanguage{french}{texte} par \textfr{texte} et
\begin{otherlanguage*}{french} \end{otherlanguage*} par \begin{fr} \end{fr}.
\babeltags peut s'appliquer à plusieurs langues : \babeltags{fr=french, de=german}.
```

Le codage d'entrée est *obligatoirement* utf8, aucun appel à `\usepackage[...]{inputenc}` n'est donc à faire. Il est recommandé d'ajouter `\usepackage{fontspec}`.

Dans les deux cas, le document sera composé avec les fontes LM ou *Latin Modern* qui sont la version moderne des fontes CM ou

Pour LuaLaTeX, il convient de faire appel aux fontes OpenType, l'exemple suivant sélectionne une police pour chaque famille (romain, sans-serif, chasse fixe) et uniformise la hauteur des minuscules :

- 
1. En fait américaine (US-english), il existe une variante `british` pour l'anglais « britannique ».
  2. Les options `frenchb` et `français` (équivalentes à `french` depuis 2004), sont conservées pour des raisons de compatibilité, mais elles *ne devraient plus être utilisées*.
  3. Là encore, le nom de la langue française est `french`, pas `frenchb` ou `français`.
  4. Rien n'empêche de remplacer `fr` par `french`, on retrouve ainsi la syntaxe de `polyglossia`.

```

\usepackage{fontspec}
\setmainfont{Erewhon}
\setsansfont[Scale=MatchLowercase]{Cabin}
\setmonofont[Scale=MatchLowercase,HyphenChar=None]{Inconsolatazi4}

```

Il est recommandé d'ajouter également `\usepackage{realscripts}` afin de profiter de vraies lettres supérieures lorsqu'elles sont disponibles.

## 2 Description de la francisation par `babel-french`

Dans un document multilingue, il y a des conventions typographiques qui changent avec la langue, comme la présence ou non d'espaces avant la ponctuation haute et d'autres, la présentation des listes, des notes de bas de page ou le retrait des premiers paragraphes des sections qui devraient s'appliquer globalement à tout le document.

`babel-french` utilise la notion de *langue principale* qui est la *dernière option* (éventuellement la seule) de la commande `\usepackage[...]{babel}`; c'est elle qui impose la présentation globale du document (listes, notes de bas de page, retrait des premiers paragraphes), les autres conventions typographiques restent locales (elles varient selon la langue utilisée). Lorsque le français n'est pas la langue principale, `babel-french` ne modifie en rien la présentation globale du document : celle-ci est imposée uniquement par la classe et les autres extensions chargées.

Lorsque le français est la langue principale, la présentation du document (ou maquette) est modifiée de la façon suivante<sup>5</sup> :

- Le premier paragraphe de chaque section est mis en retrait comme les suivants.
- Listes « `itemize` » :
  - les marqueurs traditionnels du type « • » sont remplacés par défaut par des tirets longs « — », ou par un autre marqueur choisi par l'utilisateur (voir section 3);
  - les espaces verticaux ajoutés par LaTeX entre les différents éléments d'une liste (*items*) sont supprimés.
  - la largeur des marges gauches dans les listes `itemize` est ajustée en fonction du marqueur utilisé. Le même réglage s'applique aussi aux listes `enumerate` et trois paramètres (`\listindentFB`, `\descindentFB` et `\labelwidthFB`) ont été ajoutés pour permettre d'affiner la présentation des listes `itemize`, `enumerate` et `description` (voir section 3.3).
- Par défaut les espacements verticaux de *toutes* les listes (`enumerate`, `description` mais aussi `abstract`, `quote`, `quotation`, `verse`) sont réduits.
- Les notes de bas de page sont présentées « à la française » comme ceci<sup>6</sup> au lieu de ceci<sup>7</sup>. Noter, outre la présentation différente du numéro dans la note elle-même, l'ajout de l'espace fine ajoutée avant l'appel de la première note. Le retrait des notes par rapport à la marge gauche est par défaut fixé au maximum de `\parindent` et de 1.8em, il peut être modifié en donnant la valeur voulue à `\parindentFFN` dans le préambule : `\setlength{\parindentFFN}{0mm}` par exemple. De même, le point qui suit par

5. Il est possible, pour chacun des points suivants, de revenir aux réglages standard de LaTeX, voir section 3.

6. Une note de bas de page « à la française ».

<sup>7</sup>Une note de bas de page standard (classe `article`), ça jure avec la précédente, non ?

défaut le numéro de note, ainsi que l'espace insécable qui sépare ce point du texte de la note peuvent être redéfinis : ajouter dans le préambule `\renewcommand{\dotFFN}{}`, `\renewcommand{\kernFFN}{--}` (pas de point, un tiret double sans espace après le numéro de note).

En ce qui concerne les conventions typographiques locales (dépendantes de la langue) la commande `\selectlanguage{french}` produit les effets suivants :

- Les motifs de césures françaises sont activés.
- Des espaces insécables et de taille adéquate sont ajoutés automatiquement devant la ponctuation haute (`;` `:` `!` `?`) si nécessaire<sup>8</sup>.
- La commande `\today` retourne la date en français.
- Les titres (« caption names » en anglais) sont traduits en français, ainsi la commande `\chapter` imprimera « Chapitre » au lieu de « Chapter ». Voir section 3.2 p. 12 comment modifier ces intitulés.

La commande `\selectlanguage{english}` ramène au comportement standard de LaTeX (typographie américaine).

Des commandes ont été prévues pour faciliter la saisie :

- Les guillemets français peuvent être saisis grâce à la commande `\frquote{texte}` qui affiche « *texte* » avec les espaces insécables adéquates. Il est également possible de coder `\og texte\fg{}` (ancienne syntaxe toujours valide).

Si les caractères « et » sont accessibles au clavier<sup>9</sup>, ils peuvent être utilisés pour saisir les guillemets français, `babel-french` ajoute automatique les espaces insécables : on peut coder indifféremment « a » ou «a». Il en va de même pour les guillemets simples « et ».

L'usage de `\frquote{}` est recommandé pour les citations longues (c.-à-d. s'étendant sur plus d'un paragraphe) et pour les citations imbriquées.

Pour les premières, `\frquote{}` insère automatiquement un guillemet ouvrant au début de chaque paragraphe, sauf si `EveryParGuill=close` (guillemet fermant dans ce cas) ou si `EveryParGuill=none` (aucun ajout).

Une commande `\NoEveryParQuote` permet de supprimer localement des guillemets de début de paragraphe ajoutés inconsidérément par la commande `\frquote{}` notamment dans les listes (après les labels); elle doit être utilisée dans un environnement ou un groupe pour en limiter la portée.

Pour les citations imbriquées, plusieurs présentations sont proposées selon les options choisies :

- Les citations internes sont balisées par des guillemets anglais “comme ceci” (recommandation de Aurel Ramat) sauf si `InnerGuillSingle=true`, dans ce cas les guillemets anglais sont remplacés par guillemets français simples « comme ceci » (suggestion de Jean Méron).
- Il est possible d'insérer en début de chaque ligne de la citation interne un guillemet dit « de suite », ouvrant – comme le recommande l'Imprimerie nationale – avec l'option `EveryLineGuill=open`, ou fermant (préconisé par Jean-Pierre Lacroux) avec l'option `EveryLineGuill=close`. Dans les deux cas l'option `InnerGuillSingle` est sans effet, la citation interne est toujours balisée par des guillemets chevrons « et ».

---

8. Les rares cas où l'ajout d'espace ne doit pas avoir lieu (1:2, http://, C:\, etc.) sont maintenant gérés automatiquement.

9. grâce à une touche Compose par exemple...

Lorsque les citations internes et externes se terminent en même temps, il est d'usage de supprimer le guillemet fermant de la citation interne. Pour ce faire, il suffit de coder la citation interne avec `\frquote*{}` au lieu de `\frquote{}`.

Exemple de citation imbriquée :

<p>Xavier raconte ainsi sa mésaventure :  « Au moment d'enregistrer mes bagages, l'hôtesse m'a dit tout bonnement : « Je suis désolée, il n'y a plus de place. Vous allez devoir attendre le prochain vol.  « C'est un effet de ce qu'on appelle la surréservation, ou <i>surbooking</i> en anglais. » »</p>
--

InnerGuillSingle

<p>Xavier raconte ainsi sa mésaventure :  « Au moment d'enregistrer mes bagages, l'hôtesse m'a dit tout bonnement : « Je  « suis désolée, il n'y a plus de place. Vous  « allez devoir attendre le prochain vol.  « C'est un effet de ce qu'on appelle  « la surréservation, ou <i>surbooking</i> en  « anglais. » »</p>
--

EveryLineGuill=open

Le codage est le suivant : `Xavier raconte... \frquote{Au moment... l'hôtesse m'a dit tout bonnement : \frquote*{Je suis désolée, ... en anglais.}}`

— La commande `\up` facilite la saisie des exposants en mode texte : `M\up{me}` imprime  $M^{\text{me}}$ , `1\up{er}` donne  $1^{\text{er}}$ ; on dispose aussi de `\ier \iere \iers \ieres \ieme \iemes` pour  $1^{\text{er}}$ ,  $1^{\text{re}}$ ,  $1^{\text{ers}}$ ,  $1^{\text{res}}$ ,  $2^{\text{e}}$ ,  $2^{\text{es}}$ . La commande `\up` utilise les lettres supérieures de la police lorsqu'elles sont disponibles et les simule sinon. On obtient de vraies lettres supérieures à condition de charger les extensions `fontspec` et `realscripts` et d'utiliser une police de type OpenType qui connaisse la directive « sups » (c'est le cas de la plupart d'entre-elles, il est possible de s'en assurer sous linux grâce à la commande `otfinfo -f `kpsewhich nom_police.otf``).

De plus `\up` empêche le passage en capitales des lettres supérieures dans les hauts de page par exemple.

Une variante étoilée `\up*` est prévue pour les polices qui disposent d'un jeu incomplet de lettres supérieures : la police OpenType Jenson Pro par exemple n'ont pas de « g supérieur »; en codant `M\up{gr}` on obtient  $Mg^{\text{r}}$  (Jenson) tandis que `M\up*{gr}` force l'utilisation de supérieures simulées ce qui pallie l'absence du « g supérieur », le résultat est  $Mg^{\text{gr}}$ .

— Les commandes `\bname{}` (*boxed name*) et `\bsc{}` (*boxed small caps*) facilitent la saisie des noms propres : toutes deux empêchent la coupure de leur argument, sauf au niveau du tiret pour les noms composés; la seconde imprime son argument en petites capitales, ce qui est d'usage par exemple dans les bibliographies ou les signatures.

Exemples : `\bname{Montesquieu}`, `Albert~\bsc{Camus}`.

N.B. Ces commandes ne mettent pas leur argument dans une `\mbox{}`, leur effet n'est pas aussi radical : un `\kern0pt` inhibe toute coupure du mot suivant, ce qui marche bien pour les noms simples ou composés avec un tiret et a l'avantage de préserver l'expansion éventuellement faite par `microtype`. Les patronymes à particule peuvent poser problème : coder `Jean~\bname{de La Fontaine}` n'aurait strictement aucun effet (les coupures après 'de', après 'La' ou 'Fon-taine' restent possibles), en revanche `Jean~de~La~\bname{Fontaine}` empêcherait toute coupure.

— les commandes `\primo`, `\secundo`, `\tertio` et `\quarto` peuvent être utilisées dans les énumérations; elles donnent  $1^{\circ}$ ,  $2^{\circ}$ ,  $3^{\circ}$ ,  $4^{\circ}$ . Ensuite, `\FrenchEnumerate{6}` donne  $6^{\circ}$ .

— Les abréviations de « numéro »,  $N^{\circ}$ ,  $N^{\text{os}}$ ,  $n^{\circ}$  et  $n^{\text{os}}$ , sont obtenues en tapant `\No`, `\Nos`, `\no` et `\nos`; elles incluent une espace insécable finale : coder `\no1` ou `\no 1` suffit.

— La commande `\frenchdate{année}{mois}{jour}`, qui prend trois arguments nu-

mériques obligatoires, affiche les dates en français : `\frenchdate{2001}{01}{01}` donne 1<sup>er</sup> janvier 2001, le résultat est inclus par défaut dans une boîte (`\hbox`) afin de ne jamais subir de coupure de ligne (cf. Gouriou, Ramat). En cas de difficulté il est possible de redéfinir, localement par exemple, les commandes `\FBdatebox` (`\hbox` par défaut) et `\FBdatespace` (`\space` par défaut) : `\renewcommand*\FBdatebox{\relax}` supprime la `\hbox` tandis que `\renewcommand*\FBdatespace{\~}` rend les deux espaces insécables. L'effet de ces deux commandes appliquées simultanément est d'interdire la coupure sur les espaces mais d'autoriser éventuellement celle du mois (jan-vier, dé-cembre, ...) qui est interdite par défaut.

— En mode mathématique, la virgule est toujours suivie d'une espace car elle est traitée comme un signe de ponctuation et non comme une virgule décimale<sup>10</sup>. La commande `\DecimalMathComma` supprime cette espace (mais uniquement en français), on revient au comportement standard avec `\StandardMathComma`. On peut l'utiliser dans un groupe pour limiter sa portée, sinon après une commande `\DecimalMathComma`, il est nécessaire de saisir une espace (fine) dans les listes et les intervalles par exemple  $(x, \, y)$  et  $[0, \, 1]$ .

`\DecimalMathComma` peut être placée soit dans le préambule, soit dans le corps du document *en mode texte* et dans une partie *en français*, son effet survit à un changement de langue (passage en anglais et retour en français par exemple), sauf bien sûr si elle est placée dans un groupe. Une solution alternative consiste à utiliser l'extension `icomma`.

— La commande `\nombre`, destinée à formater automatiquement les nombres entiers ou décimaux par tranches de trois chiffres séparées par des espaces en français et par des virgules (usage anglo-saxon), fait désormais appel à la commande `\numprint` de l'extension du même nom. Lors du premier appel à la commande `\nombre`, un message est affiché dans le fichier `.log` indiquant comment charger `numprint`. Le chargement de `numprint` n'est pas fait par `babel-french` à cause du risque de conflit d'options. Il doit se faire *après* `babel`. Les utilisateurs devraient s'habituer progressivement à utiliser `\numprint` (ou son raccourci `\np`) à la place de `\nombre`.

En ajoutant `\usepackage{xspace}` dans le préambule, les espaces suivant les commandes `\ier, \dots, \ieres, \ieme, \iemes, \fg` et `\dots` sont respectés sans avoir à les forcer par des `{}` ou des `\_`. Le recours à l'extension `xspace` est cependant une *fausse simplification* car *certaines* commandes sont affectées mais *pas toutes*. Il me paraît plus simple d'éviter le recours à `xspace` et de gérer les espaces soi-même : « les 1<sup>ier</sup>, 2 et 3<sup>~</sup>mai » ou « le 1<sup>ier</sup>~mai » (espace insécable).

### 3 Personnalisation

La commande `\frenchsetup{}`, appelée précédemment `\frenchbsetup{}`<sup>11</sup>, est à placer dans le préambule de chaque document après le chargement de Babel; elle permet de personnaliser le comportement de `babel-french` grâce au large choix parmi des options disponibles. La syntaxe est celle de l'extension `keyval`, largement utilisée par d'autres extensions comme `geometry` ou `hyperref`.

Le recours à un fichier de configuration `frenchb.cfg` n'est pas possible.

10. Une virgule décimale peut toujours être codée `{,}` en mode math.

11. Ce dernier nom sera gardé comme alias par souci de compatibilité, le nouveau devrait être préféré depuis l'abandon du nom `frenchb` au profit de `babel-french` pour désigner l'option `french` de `babel`.

### 3.1 `\frenchsetup{< options >}`

La commande `\frenchsetup{ShowOptions}` affiche dans le fichier `.log` la liste des options disponibles, nous allons parcourir cette liste et expliquer l'effet de chacune d'elles. Dans le cas d'une option booléenne, la mention `=true` peut être omise : `\frenchsetup{ShowOptions}` est équivalent à `\frenchsetup{ShowOptions=true}`.

Dans la liste ci-dessous, l'option activée par défaut est indiquée entre parenthèses, éventuellement suivie d'un étoile. L'étoile indique que la valeur par défaut correspond au cas où le français *est la langue principale* (voir section 2, page 4), et que cette valeur est inversée sinon.

La liste étant longue, les options sont regroupées par thèmes.

#### Inventaire des options

**ShowOptions=true (false)** affiche dans le fichier `.log` toutes les options disponibles, ce qui permet de retrouver leurs noms facilement.

#### Maquette générale

**StandardLayout=true (false\*)** supprime toute action de `babel-french` sur la maquette dans le cas où le français est la langue principale : retour aux listes standard, pas de retrait des 1<sup>ers</sup> paragraphes des sections, notes de bas de page standard. Lorsque le français n'est pas la langue principale, elle est sans effet.

**IndentFirst=false (true\*)** ; par défaut, `babel-french` applique un retrait identique pour tous les paragraphes (de largeur `\parindent`), y compris le premier de chaque section, ce qui est conforme à l'usage français. Avec `IndentFirst=false` le retrait du premier paragraphe de chaque section est supprimé, comme c'est l'usage en anglais, soit dans tout le document si le français est la langue principale, soit seulement en français.

**PartNameFull=false (true\*)** ; par défaut `babel-french` numérote les parties créées par la commande `\part` en « Première partie », « Deuxième partie », etc. Ceci fonctionne bien pour la plupart des classes (classes standard, classes koma-script, memoir) mais pas pour les classes AMS qui redéfinissent la commande `\part`. L'option `PartNameFull=false` permet de revenir à une numérotation des parties plus standard – « Partie I », « Partie II », etc. –, ce qui supprime les risques de mauvais affichage du genre « Première partie 1 » notamment dans la table des matières.

**TocPartNameFull=false (true\*)** ; les numéros de parties sont également affichés sous la forme « Première partie », « Deuxième partie », avec les classes koma-script et `memoir` uniquement<sup>12</sup>. Il est possible de supprimer ce comportement en mettant cette option à `false`.

#### Présentation des listes

**ListItemsAsPar=true (false)** ; mettre cette option à `true` est recommandé à ceux qui souhaitent une présentation des listes conforme aux usages en vigueur à l'Imprimerie nationale, voir p. 15 pour une comparaison avec la disposition par défaut.

---

12. Les classes standard ne fournissent pas de *hook* pour personnaliser la table des matières.



**StandardListSpacing=true (false\*)** ; par défaut `babel-french` modifie (en général en les réduisant, sauf pour les classes SMF) les espaces verticaux<sup>13</sup> dans *toutes les listes* produites à partir de l'environnement `list`, en particulier les listes `enumerate`, `itemize` et `description` mais aussi `abstract`, `quote`, `quotation`, `verse`... `StandardListSpacing=true` permet de revenir aux réglages standard (ceux de la classe utilisée), elle remplace l'ancienne option `ReduceListSpacing` en l'inversant : `StandardListSpacing=true` équivaut à `ReduceListSpacing=false` qui pouvait être trompeuse dans le cas des classes SMF mais qui fonctionne toujours pour ne pas compromettre la compilation d'anciens documents.

**StandardItemizeEnv=true (false\*)** ; par défaut l'environnement `itemize` est redéfini pour qu'aucun espace vertical ne soit ajouté à l'interligne standard entre les éléments d'une liste `itemize` et pour adapter `\labelwidth` au marqueur utilisé.

L'option `StandardItemizeEnv=true` empêche cette redéfinition, ce qui est nécessaire en cas de conflit avec une classe ou une extension qui redéfinit aussi l'environnement `itemize` (`enumitem` ou `paralist` par exemple<sup>14</sup>). L'interligne entre les éléments des listes `itemize` est alors légèrement augmenté si `StandardListSpacing=false` ou l'est nettement plus (on revient au réglage de base de la classe utilisée) si `StandardListSpacing=true`.

Les utilisateurs d'`enumitem` qui souhaitent obtenir pour leurs listes une présentation à la française, trouveront comment faire page 15.

**StandardEnumerateEnv=true (false\*)** ; depuis la version 2.6, `babel-french` redéfinit également les environnements `enumerate` et `description` pour que leurs marges soient les mêmes que celles des listes `itemize`.

`StandardEnumerateEnv=true` empêche cette redéfinition, ce qui est nécessaire en cas de conflit avec une classe ou une extension qui redéfinit aussi l'environnement `enumerate` (`enumerate`, `enumitem` ou `paralist` par exemple<sup>15</sup>).

**StandardItemLabels=true (false\*)** restitue aux marqueurs des listes `itemize` les valeurs standard attribuées par la classe de document ou les extensions utilisées.

**ItemLabels=\textbullet, \textendash, \ding43, ... (\textemdash\*)** ; option qui permet de choisir le marqueur utilisé dans les listes `itemize` en français (*sauf* bien sûr si `StandardItemLabels=true`). Noter que `\ding{43}` suppose que l'extension `pifont` soit chargée. Cette option affecte tous les niveaux de la liste. Les quatre options suivantes fonctionnent de même mais n'affectent elles qu'un niveau chacune :

**ItemLabeli=\textbullet, \textendash, \ding43, ... (\textemdash\*)**

**ItemLabelii=\textbullet, \textendash, \ding43, ... (\textemdash\*)**

**ItemLabeliii=\textbullet, \textendash, \ding43, ... (\textemdash\*)**

**ItemLabeliv=\textbullet, \textendash, \ding43, ... (\textemdash\*)**

**StandardLists=true (false\*)** ; supprime toute action de `babel-french` sur les listes, elle équivaut aux quatre options `StandardItemLabels=true`, `StandardItemizeEnv=true`, `StandardEnumerateEnv=true` et `StandardListSpacing=true`. Lors de l'utilisation d'une classe ou d'une extension qui modifie la présentation des listes, il peut y avoir conflit avec `babel-french`; dans ce cas l'option `StandardLists` (ou éventuellement l'une ou l'autre des

---

13. Il s'agit de `\itemsep`, `\parsep`, `\topsep` et `\partopsep`.

14. `babel-french` met automatiquement le drapeau `StandardItemizeEnv` à `true` lorsque l'une des extensions `enumitem` ou `paralist` est chargée.

15. `babel-french` met automatiquement le drapeau `StandardEnumerateEnv` à `true` lorsque l'une des extensions `enumerate`, `enumitem` ou `paralist` est chargée.

quatre sous-options plus ciblées qu'elle regroupe) devrait régler le problème<sup>16</sup>.

## Présentation des notes de bas de page

**FrenchFootnotes=false (true\*)** fait revenir à la présentation standard des notes de page, telle que définie par la classe ou les extensions utilisées. Cette option, `true` ou `false`, affecte la totalité du document, les notes numérotées avec des symboles (commande `\thanks{}` de `\maketitle{}`) ou avec des lettres (environnement `minipage`) gardent elles toujours la présentation standard.

**AutoSpaceFootnotes=false (true\*)** supprime l'espace fine insécable ajoutée par défaut avant l'appel de chaque note dans le texte courant. Cette option affecte la totalité du document. L'espace éventuellement ajoutée peut être ajustée, la commande par défaut est : `\newcommand*{\FBfnmarkspace}{\kern .5\fontdimen2\font}`

## Ponctuation haute

**AutoSpacePunctuation=false (true)** ; basculer cette option à `false` est *vivement déconseillé*. Cette option n'est utile qu'avec les moteurs pdfTeX et XeTeX.

Le moteur LuaTeX permet un contrôle fin de l'ajout d'espace devant le deux-points : si ce caractère n'est pas précédé d'un espace et que celui qui le suit est de type *glyphe* (lettre, chiffre, /, \, etc.) aucune espace n'est ajoutée en sortie, ce qui évite les espaces parasites dans les URLs (`ftp://monsite`), les chemins MSDOS (`C:\`), les horaires (`10:55`) et les ratios (`1:1`).

Si l'utilisateur code une espace insécable U+00A0 (*pas* ~) ou une fine insécable U+202F, ces caractères sont recopiés tels quels en sortie, sans ajout d'espace par `babel-french`.

Noter que l'insertion automatique d'espaces peut être inhibée localement grâce à la commande `\NoAutoSpacing` utilisée dans un groupe comme ceci : `{\NoAutoSpacing xx: yy}`.

**ThinColonSpace=true (false)** ; par défaut l'espace placée avant le caractère ':' est une espace-mot insécable, cette option la remplace par une espace fine insécable. Certains auteurs font ce choix pour que les espaces précédant les quatre signes de ponctuation haute soient identiques. Le choix par défaut correspond à la maquette de l'Imprimerie nationale, le Guide du typographe (ex-roman) préconise lui l'espace fine devant un ':'.

Il est possible de redéfinir complètement la taille des espaces précédant la ponctuation haute (ou les guillemets français) en utilisant dans le préambule la commande `\FBsetspaces` qui admet quatre arguments obligatoires : le premier précise le type d'espace à redéfinir, il doit être `colon` (pour ':'), `thin` (pour ';' '!' et '?') ou `guill` pour les guillemets chevrons), les trois suivants sont des nombres décimaux qui définissent la largeur (*width*), l'extensibilité (*stretch*) et la compressibilité (*shrink*) de l'espace exprimées en *fontdimen*.

Exemples : les valeurs `{1}{1}{1}` correspondent à l'espace-mot justifiante (utilisée par défaut devant le deux-points), `{0.5}{0}{0}` à l'espace fine (utilisée par défaut devant ';' '!' et '?', noter son absence d'élasticité) et `{0.8}{0.3}{0.8}` à l'espace utilisée par défaut à l'intérieur des guillemets.

`\FBsetspaces{colon}{0.5}{0}{0}` est équivalent à l'option `ThinColonSpace` (vaut pour le français et le canadien).

---

16. L'option `StandardLists` est automatiquement activée avec la classe `beamer`.

`\FBsetspace{colon}{1}{0}{0}` met une espace-mot sans élasticité devant les ‘:’.

**OriginalTypewriter=true (false)** ; par défaut, lorsqu’une police à espacement fixe est utilisée (`\texttt{}`, mode verbatim, listings, etc.) `babel-french`, n’ajoute jamais d’espace avant la ponctuation haute ni après ‘«’ ou ‘<’, ni avant ‘»’ ou ‘>’. De plus, les espaces présents dans le texte source ne sont pas modifiés, par exemple `\texttt{X != Y}` produit `X != Y` (pas d’espace fine avant le ‘!’). Mettre cette option à `true` supprime ce comportement.

**UnicodeNoBreakSpaces=true (false)** ; lorsqu’on la met à `true` les espaces insécables ajoutées par `babel-french` dans le fichier PDF de sortie (ponctuation haute, guillemets, etc.) sont codées sous forme de caractères Unicode (U+A0 ou U+202F fine) au lieu des pénalités et ressorts de type *glues*. Ceci devrait préserver ces espaces lors de la conversion en HTML des fichiers produits par LuaLaTeX, malheureusement la version actuelle de `pdftotext` ne les restitue pas correctement.

En revanche, `lwrap` (v. 0.37 et suivantes) est totalement compatible avec `babel-french` pour la conversion en HTML des fichiers PDF.

## Guillemets

**og=«, fg=»** ; cette option — qui se justifiait en présence de codages d’entrée disparates — n’a plus d’effet, elle affiche seulement un avertissement. `babel-french` gère automatiquement l’espacement des guillemets français.

Les espaces insécables ne sont ajoutées que lorsque la langue courante est le français : en allemand, le codage (`»Auf Deutsch«`) produit (`»Auf Deutsch«`).

**INGuillSpace=true (false)** ; force `babel-french` à mettre une espace-mot justifiante insécable après les guillemets ouvrants et avant les guillemets fermants comme le préconise l’Imprimerie nationale. Par défaut les espaces insérées par `babel-french` sont légèrement plus étroites et moins extensibles qu’une espace-mot.

Cette option est équivalente à `\FBsetspace{guill}{1}{1}{1}`, voir p. 10.

Les options suivantes n’affectent que les citations codées *intégralement* avec `\frquote{}`.

**InnerGuillSingle=true (false)** ; si `InnerGuillSingle=true` les citations internes sont balisées comme `< ceci >`, sinon comme `“ cela ”`.

**EveryParGuill=open, close, none (open)** ; selon sa valeur, cette option ajoute un guillemet ouvrant (`«`), fermant (`»`) ou rien au début de chaque paragraphe inclus dans une citation de premier rang codée avec `\frquote{}`.

Lorsque `InnerGuillSingle=true`, cette option est également prise en compte pour les citations de second rang (internes) : selon sa valeur, un guillemet simple ouvrant (`<`) ou fermant (`>`) ou rien, est ajouté à chaque début de paragraphe.

Lorsque `InnerGuillSingle=false`, rien n’est ajouté en début de paragraphe dans les citations de second rang.

**EveryLineGuill=open, close, none (none)** ; selon sa valeur, cette option ajoute un guillemet ouvrant (`«`), fermant (`»`) ou rien, au début de chaque ligne d’une citation de second rang. Notez que les citations de premier et de second rang doivent être toutes deux codées avec `\frquote{}`. Lorsque `EveryLineGuill=open` ou `close` la citation interne est toujours balisée par des guillemets français doubles `« et »`, même si `InnerGuillSingle=true`.

## Présentation des nombres

**ThinSpaceInFrenchNumbers=true (false)** remplace le séparateur des milliers utilisé en français par la commande `\numprint{}` (ou son alias `\nombre{}`) pour le formatage des nombres, par une espace fine (par défaut c'est une espace mot insécable et sans élasticité en français). Cette option n'a d'effet que si l'extension `numprint` est chargée (*après babel*) avec l'option `autolanguage`; sans elle, `numprint` formate les nombres indépendamment de la langue courante et le séparateur des milliers est par défaut l'espace fine.

## Légendes de figures et tables

**SmallCapsFigTabCaptions=false (true\*)** ; si cette option est mise à `false`, le recours aux petites capitales dans les intitulés des légendes de figures et tables est supprimé, on obtient « Figure » et « Table » au lieu de « FIGURE » et « TABLE ». Noter que le même résultat peut être obtenu en définissant `\FBfigtabshape` comme `\relax` avant le chargement de `babel`.

**CustomiseFigTabCaptions=true (false)** ; si cette option est mise à `true`, le séparateur par défaut (deux-points) est remplacé par ‘-’ dans les légendes des figures et des tables, ceci pour toutes les langues. Le séparateur ‘:’ est maintenant toujours précédé d'une espace insécable en français, il n'y a donc plus aucun intérêt à forcer cette option à `true`.

## Divers : lettres supérieures et « warnings »

**LowercaseSuperscripts=false (true)** rend possible d'avoir des lettres supérieures en capitales (est-ce bien utile?). Par défaut, la commande `\up` empêche le passage en capitales des lettres supérieures dans les hauts de pages par exemple.

**SuppressWarning=true (false)** peut être utilisée pour supprimer les avertissements non essentiels émis par `babel-french`.

## Ordre des options

Il faut se souvenir que les options sont prises en compte dans l'ordre où elles sont écrites dans la commande `\frenchsetup{}`.

Exemple : un utilisateur souhaitant que `babel-french` ne touche pas à la présentation des listes ni à celle des notes de bas de page mais ajoute un renforcement au début des 1<sup>ers</sup> paragraphes de section peut faire `\frenchsetup{StandardLayout, IndentFirst}`. S'il choisissait l'ordre inverse l'option `IndentFirst` serait annulée par `StandardLayout`.

Cet utilisateur obtiendrait également le résultat souhaité en codant `\frenchsetup{StandardLists, FrenchFootnotes=false}` là, l'ordre est indifférent car les options sont indépendantes.

## 3.2 Traduction des intitulés (« caption names » en anglais)

Voici la liste des traductions proposées par `babel-french` :

Commande Babel	Commande pour le français	Intitulé par défaut en français
<code>\abstractname</code>	<code>\frenchabstractname</code>	Résumé
<code>\bibname</code>	<code>\frenchbibname</code>	Bibliographie
<code>\refname</code>	<code>\frenchrefname</code>	Références
<code>\prefacename</code>	<code>\frenchprefacename</code>	Préface
<code>\chaptername</code>	<code>\frenchchaptername</code>	Chapitre
<code>\appendixname</code>	<code>\frenchappendixname</code>	Annexe
<code>\contentsname</code>	<code>\frenchcontentsname</code>	Table des matières
<code>\listfigurename</code>	<code>\frenchlistfigurename</code>	Table des figures
<code>\listtablename</code>	<code>\frenchlisttablename</code>	Liste des tableaux
<code>\indexname</code>	<code>\frenchindexname</code>	Index
<code>\glossaryname</code>	<code>\frenchglossaryname</code>	Glossaire
<code>\figurename</code>	<code>\frenchfigurename</code>	Figure
<code>\tablename</code>	<code>\frenchtablename</code>	Table
<code>\partname</code>	<code>\frenchpartname</code>	Première partie, Deuxième partie...
<code>\enclname</code>	<code>\frenchenclname</code>	P. J.
<code>\ccname</code>	<code>\frenchccname</code>	Copie à
<code>\headtoname</code>	<code>\frenchheadtoname</code>	< <i>vide</i> >
<code>\pagename</code>	<code>\frenchpagename</code>	page
<code>\seename</code>	<code>\frenchseename</code>	voir
<code>\alsoname</code>	<code>\frenchalsoname</code>	voir aussi
<code>\proofname</code>	<code>\frenchproofname</code>	Démonstration

Il est facile de modifier ces traductions : pour remplacer « Démonstration » par « Preuve » (avec `amsthm`) ajouter dans le préambule `\def\frenchproofname{Preuve}`<sup>17</sup>, noter que l'ancienne syntaxe `\addto\captionsfrench{\def\proofname{Preuve}}` fonctionne toujours.

La modification des noms de parties est plus complexe : si on préfère « Partie I », « Partie II » à « Première partie », « Deuxième partie », il suffit de redéfinir `\frenchpartname` comme ceci `\def\frenchpartname{Partie}`<sup>17</sup>; utiliser l'option `PartNameFull=false` dans `\frenchsetup{}` aboutit au même résultat. Si en revanche on veut que `\part{}` produise par exemple « Première phase », « Deuxième phase », il faut redéfinir `\frenchpartnameord` et non `\frenchpartname` : `\def\frenchpartnameord{phase}`<sup>17</sup>. Le remplacement de « Deuxième » par « Seconde » est possible en codant `\def\frenchpartsecond{Seconde}`<sup>17</sup> dans le préambule. De même il est possible de remplacer « Première » par « Premier » en codant `\def\frenchpartfirst{Premier}`<sup>17</sup>.

On remarquera que `\figurename` et `\tablename` sont affichés en petites capitales en français. Il serait préférable, notamment dans un document multilingue, de s'en tenir à la traduction pure en actionnant l'option `SmallCapsFigTabCaptions=false` (p. 12).

### 3.3 Présentation des listes

Voici la présentation par défaut<sup>18</sup> pour les listes `itemize` (tiret cadratin et alignement des tirets de premier niveau sur le retrait (`\parindent`) de début de paragraphe) :

17. Les puristes peuvent remplacer `\def` par `\renewcommand*` s'ils le souhaitent.

18. L'option `ListItemsAsPar` permet une autre présentation, voir ci-dessous p. 15.

```
<= Marge gauche
Paragraphe précédant la liste
— Premier élément sur plus
d'une ligne...
— Un élément niveau 2,
— Second élément niveau 2,
— Second élément...
```

Ceux qui préfèrent le tiret demi-cadratin (–) comme marqueur peuvent ajouter l'option `\frenchsetup{ItemLabels=\textendash}`, d'autres options existent voir section 3.1.

`babel-french` applique aux listes `enumerate` et `description` les mêmes retraits horizontaux qu'aux listes `itemize`, tous sont donc basés sur la largeur du marqueur choisi, `\textendash` par défaut, `\textendash`, ou autre.

Trois paramètres dimensionnels permettent d'ajuster la présentation des listes `itemize`, `enumerate` et `description` :

`\listindentFB` dont la valeur par défaut est `\parindent` si `\parindent` est non nul et `1.5em` (valeur standard de `\parindent`) sinon; `\listindentFB` permet modifier la marge gauche du premier niveau de liste (ce qui déplace aussi les listes incluses).

Exemple : en ajoutant `\setlength{\listindentFB}{0pt}` dans le préambule, les étiquettes de toutes les listes de premier niveau colleront à la marge gauche au lieu d'être décalées vers la droite.

```
<= Marge gauche
Paragraphe précédant la liste
— Premier élément sur plus
d'une ligne...
— Un élément niveau 2,
— Second élément niveau 2,
— Second élément...
```

`\parindent=0pt`

```
<= Marge gauche
Paragraphe précédant la liste
— Premier élément sur plus d'une
ligne...
— Un élément niveau 2,
— Second élément niveau 2,
— Second élément...
```

`\parindent=0pt et \listindentFB=0pt`

`\labelwidthFB` dont la valeur par défaut est la largeur de `\FrenchLabelItem` (c.-à.d. `\textendash` sauf changement décidé par l'utilisateur); il est possible de fixer la valeur de `\labelwidthFB` niveau par niveau, voir le second exemple ci-dessous.

```
<= Marge gauche
Paragraphe précédant la liste
1. Premier élément sur plus
d'une ligne...
(a) Un élément niveau 2,
(b) Second élément niveau 2,
2. Second élément...
```

Présentation par défaut<sup>1</sup>

```
<= Marge gauche
Paragraphe précédant la liste
1. Premier élément sur plus d'une
ligne...
(a) Un élément niveau 2,
(b) Second élément niveau 2,
2. Second élément...
```

`\settowidth{\labelwidthFB}{(a)}`<sup>2</sup>

<sup>1</sup>Remarquer l'alignement vertical avec l'exemple ci-dessus.

<sup>2</sup>Commande à placer juste avant le second environnement `enumerate`.

`\descindentFB` dont la valeur par défaut est `\listindentFB` permet de traiter les listes de type `description` différemment des autres : pour que les étiquettes des seules listes `description` collent à la marge gauche, il suffit d'ajouter dans le préambule `\setlength{\descindentFB}{0pt}`.

<= Marge gauche  
Paragraphe précédant la liste

**Premier** élément occupant plus d'une ligne...

- Un élément niveau 2,
- Second élément niveau 2,

**Second** élément...

`\parindent=0pt`

<= Marge gauche  
Paragraphe précédant la liste

**Premier** élément occupant plus d'une ligne...

- Un élément niveau 2,
- Second élément niveau 2,

**Second** élément...

`\parindent=0pt et \descindentFB=0pt`

L'option `ListItemsAsPar` propose une présentation plus conforme aux usages typographiques français, voir par exemple ce qui est fait dans le « Lexique des règles typographiques en usage à l'Imprimerie nationale ». Les éléments des listes sont alors présentés comme des paragraphes, les marqueurs (tirets, etc.) étant placés en retrait de la marge gauche ; dans la présentation par défaut les étiquettes sont placées en saillie dans la marge gauche et celle-ci est augmentée comme le montre l'exemple ci-dessous.

Noter que toute cette documentation a été compilée avec l'option `ListItemsAsPar` dont l'effet est bien visible dans les sections 2 (listes `itemize`) et 3 (listes `description`).

<= Marge gauche  
Paragraphe précédant la liste et qui s'étend sur deux lignes.

- Premier élément qui se prolonge sur plusieurs...
- Un élément niveau 2,
- Second élément niveau 2 sur plus d'une ligne...
- Second élément...

Présentation par défaut

<= Marge gauche  
Paragraphe précédant la liste et qui s'étend sur deux lignes.

- Premier élément qui se prolonge sur plusieurs lignes...
- Un élément niveau 2,
- Second élément niveau 2 sur plus d'une ligne...
- Second élément...

`ListItemsAsPar=true`

Enfin, pour ceux qui voudraient bénéficier des facilités offertes par l'extension `enumitem` et conserver une présentation des listes similaire à celle obtenue par défaut avec `babel-french`, je propose les quelques lignes de code suivantes à copier-coller dans le préambule :

```
\usepackage{enumitem}
\newlength\mylabelwidth
\newcommand*\mylabel{\textendash} % ou \textendash (tiret plus court)
\settowidth{\mylabelwidth}{\mylabel}
\setlist[itemize]{label=\mylabel, nosep}
\setlist[1]{labelindent=\parindent}
\setlist{labelwidth=\mylabelwidth, leftmargin=!,
itemsep=0.4ex plus 0.2ex minus 0.2ex,
parsep=0.4ex plus 0.2ex minus 0.2ex,
```

```
topsep=0.8ex plus 0.4ex minus 0.4ex,
partopsep=0.4ex plus 0.2ex minus 0.2ex}
```

Pour une présentation correspondant à l'option `ListItemsAsPar=true` on aurait :

```
\usepackage{enumitem}
\newlength\mylabelwidth
\newlength\myitemindent
\newcommand*\mylabel{\textemdash} % ou \textendash (tiret plus court)
\settowidth{\mylabelwidth}{\mylabel}
\setlength{\myitemindent}{\parindent}
\addtolength{\myitemindent}{\mylabelwidth}
\addtolength{\myitemindent}{\labelsep}
\setlist[itemize]{label=\mylabel, nosep}
\setlist[1]{leftmargin=0pt}
\setlist{leftmargin=\parindent, itemindent=\myitemindent,
itemsep=0.4ex plus 0.2ex minus 0.2ex,
parsep=0.4ex plus 0.2ex minus 0.2ex,
topsep=0.8ex plus 0.4ex minus 0.4ex,
partopsep=0.4ex plus 0.2ex minus 0.2ex}
```

## 4 Changements entre les versions 4.0e et 3.6

### 4.1 Quoi de neuf en version 4.0?

`babel-french` est maintenant scindée en deux parties, une partie « historique » (fichier `frenchb3.dtx`) réservée aux moteurs anciens (TeX, pdfTeX et XeTeX) qui restera figée — sauf corrections éventuelles de bogues –, l'autre « moderne » destinée à Lua(La)TeX (fichier `frenchb.dtx`) qui continuera à évoluer. Cette scission a permis de simplifier le code de `frenchb.dtx` et de le débarrasser de vieilleries devenues inutiles avec LuaTeX.

Certaines options ont été modifiées, supprimées ou ajoutées :

- `AutoPunctuation=false` n'a plus aucune raison d'être utilisée, LuaTeX permettant de gérer les cas exceptionnels (`2:1`, `\http://`, `C:\`, `!!`, etc.) où aucune espace ne doit être ajoutée devant la ponctuation haute. Si cette option est forcée à `false`, un message dans le fichier `.log` déconseille explicitement ce choix.

- `\frenchsetup{og=«, fg=»}` n'a plus aucun effet<sup>19</sup>, elle génère juste un avertissement dans le fichier `.log`. Les guillemets chevrons « et » génèrent automatiquement les espaces adéquates, il en va de même désormais des guillemets simples < et >. Il est possible de débrayer localement l'ajout d'espaces grâce à la commande `{\NoAutospacing}`.

- L'option `CustomiseFigTabCaptions` a été inversée, elle est maintenant `false` par défaut, ce qui revient à accepter la présentation proposée par la classe et les extensions chargées. Motivation : le deux-points est toujours affiché correctement dans les légendes de figures ou de tableaux.

Lorsque l'option `CustomiseFigTabCaptions` est forcée à `true`, un message déconseillant ce choix est affiché dans le fichier `.log`; le séparateur (`:`) est néanmoins remplacé par un tiret

19. Elle ne se justifiait qu'en présence de codages d'entrée variés.



demi-cadratin. La personnalisation des légendes devrait être faite soit par la classe, soit par l'extension `caption.sty`, exemples :

— avec la classe `memoir` ajouter :

```
\captiondelim{\space\textendash\space}
```

— avec les classes koma-script ajouter :

```
\renewcommand{\captionformat}{\space\textendash\space}
```

— avec la classe `beamer` ajouter :

```
\setbeamertemplate{caption label separator}[endash]
```

— et avec les classes `article`, `book` ou `report`, charger `caption.sty` comme ceci :

```
\usepackage[labelsep=endash]{caption}
```

— Les options `OldFigTabCaptions`, `ListOldLayout` et `GlobalLayoutFrench` sont supprimées car elles sont devenues inutiles : elles ne servaient qu'à reproduire de (très) vieux comportements de `babel-french`.

— Une nouvelle option `TocPartFullName` a été ajoutée sur une suggestion de Julien Labbé que je remercie. Elle permet d'afficher dans la table des matières les parties numérotées sous la forme « Première partie », « Deuxième partie », etc. Ceci ne fonctionne que pour les classes `memoir` et koma-script, les classes standard ne fournissant pas de *hook* permettant de personnaliser la table des matières. La commande fournissant ces chaînes s'appelle `\FBtocpartname{< Romannum >}`, elle prend en argument le numéro de partie en chiffres romains et retourne la chaîne correspondante. Il est possible d'y ajouter un séparateur (vide par défaut) : `\renewcommand*{\FBtocpartsep}{. }` ajoute un point final suivi d'une espace.

Une nouvelle fonction `euphonic_t` a été intégrée dans le code `lua` : elle empêche les coupures sur le second tiret d'expressions composées comme « va-t-on », « semble-t-il », « dira-t-elle », etc. (problème signalé par Thomas Savary).

**Note sur le projet « PDF tagging »** : ce projet de l'équipe LaTeX nécessite une redéfinition complète des listes basée sur des « templates ». Cette redéfinition n'est pas encore complètement finalisée et nécessitera quand elle le sera une réécriture de la gestion des listes par `babel-french`. Actuellement, la personnalisation des listes est désactivée et un message l'annonce dans le fichier `.log` dès que les nouvelles listes sont utilisées. C'est le cas si l'utilisateur ajoute la ligne

```
\DocumentMetadata{testphase=latest} ou \DocumentMetadata{tagging=on}
```

La version 4.0b met à profit la nouvelle interface, lorsqu'elle est disponible, pour les notes de bas de page. Ceci devrait corriger le problème 932.

Noter aussi que si une langue est déclarée dans `\DocumentMetadata{}`, elle devient automatiquement pour `babel` la langue principale du document, il est donc nécessaire d'assurer la cohérence des déclarations, ainsi le présent document contient

```
\DocumentMetadata{pdfstandard=UA-2, pdfversion=2.0, lang=fr}
\documentclass[a4paper, 12pt, german, british, french]{article}
\usepackage{babel}
```

## 4.2 Quoi de neuf en version 3.7?

Le support de la variante `acadian`<sup>20</sup>, est abandonné : `\usepackage[acadian]{babel}` émet désormais un avertissement et traite la langue `acadian` exactement comme si l'utilisateur avait codé `\usepackage[french]{babel}`.

## 4.3 Quoi de neuf en version 3.6?

Un bug affectant l'utilisation de la commande `\NoAutoSpacing` dans les signets `hyperref` a été corrigé en version 3.6b.

La version 3.6a n'a plus recours à l'extension `keyval` pour gérer les options, les commandes internes du noyau LaTeX (`l3keys`) `\DeclareKeys{}` et `\SetKeys{}` sont utilisées.

L'espace fine ajoutée avant l'appel des notes de bas de pages est maintenant personnalisable (suggestion de Thomas Savary); le nom de la commande est `\FBfnmarkspace{}`, c'est une *vraie* espace fine (demie espace-mot de la police courante) contrairement à la commande LaTeX `\,` alias `\thinspace`<sup>21</sup> précédemment utilisée qui chasse toujours 1/6em. Les deux définitions coïncident pour les polices dont l'espace-mot vaut 1/3em, (cas le plus courant), l'écart est infime pour celles qui chassent moins : il est par exemple de  $(1/6 - 1/8)12 = 0,5\text{pt}$  en 12pt pour une police dont l'espace-mot chasse 1/4em. Il est toujours possible de revenir à l'ancien comportement en codant `\renewcommand*{\FBfnmarkspace}{\,}`.

## 5 Problèmes de césures

Pour vérifier que votre format LaTeX fonctionne correctement au niveau des césures, au moins en français et en anglais, téléchargez le fichier de test <http://daniel.flipo.free.fr/frenchb/frenchb-cesures.tex> et suivez les instructions figurant en début de fichier.

Si les résultats du test ne sont pas corrects, vérifiez tout au début du fichier `.log` dans la ligne commençant par le mot « Babel », si « french » figure bien dans la liste des langues disponibles.

Si ce n'est pas le cas, installez le paquet appelé « collection-langfrench » (TeXLive, MacTeX) ou « texlive-lang-french » (distributions Linux) ou similaire. Si vous travaillez avec « tlmgr », les formats sont refaits automatiquement, sinon il faut les faire à la main (la procédure dépend de votre installation).

Recompilez le fichier `frenchb-cesures.tex`, les résultats devraient être corrects, contactez-moi par courriel si ce n'était pas le cas.

## 6 Incompatibilités connues et remèdes

La liste suivante ne prétend pas être exhaustive, n'hésitez pas à me signaler les incompatibilités que vous rencontrez afin qu'elles puissent figurer dans cette liste.

---

20. Et de son synonyme `canadien`

21. Elles sont définies pour fonctionner aussi en mode mathématique.

— Les extensions `caption`, `subcaption`, `floatrow` doivent impérativement être chargées *après* `babel`. En revanche `beamerarticle` doit être chargée *avant* `babel`.

À chaque fois que le bon ordre de chargement n'est pas respecté, un avertissement le signale dans le fichier `.log`.

— `babel-french` modifie la présentation des listes ce qui peut perturber les classes ou extensions qui veulent également le faire. `babel-french` s'efface automatiquement lorsqu'une des extensions `enumitem`, `enumerate` ou `paralist` est chargée. Pour les autres, ou si on veut soi-même agir sur la présentation des listes, il convient de débrayer l'action de `babel-french` en utilisant la commande `\frenchsetup{}` avec les options adéquates (cf. section 3.1) dans le préambule du document (après le chargement de Babel).

— Certaines classes (`amsbook`, `smfbook`, `beamer`, etc.) redéfinissent `\part`, ce qui peut conduire à des titres du genre « Première partie I ». La parade consiste à ajouter l'option `PartNameFull=false` dans `\frenchsetup{}`.

— L'option `multiple` de l'extension `footmisc` insère normalement une virgule entre les appels de notes multiples. Pour que ce mécanisme fonctionne avec `babel-french` il faut ajouter l'option `\frenchsetup{AutoSpaceFootnotes=false}`, sinon `babel-french` remplace la virgule par une espace fine.

— Dans une commande `\index{}`, les guillemets français doivent obligatoirement être codés sous la forme `\frquote{}` (ou `\og`, `\fg`) : `\index{\frquote{toto}}` donne le résultat attendu, tandis que `\index{«toto»}` provoque un mauvais classement de l'entrée «toto» dans l'index.

## 7 Bibliographie

Ce qui suit ne concerne *pas* les bibliographies faites « à la main » dans l'environnement `thebibliography`, mais celles créées à partir d'un ou plusieurs fichiers `.bib` avec `BIBTEX` ou mieux avec Biblatex/Biber.

### 7.1 Bibliographie avec `BIBTEX`

`BIBTEX` est obsolète, je conseille *vraiment* de passer à Biblatex/Biber (voir section suivante). Néanmoins, voici quelques indications concernant la francisation des bibliographies créées avec `BIBTEX` :

Certains champs (les dates notamment) et certains mots-clés figurant dans les fichiers `.bib` (les « *and* » des listes d'auteurs par exemple) devraient pouvoir être affichés différemment selon le contexte (les « *and* » du fichier `.bib` devraient pouvoir être transcrits en « *et* » dans le fichier `.bbl` pour les références en français). Babel n'opère pas au niveau `BIBTEX`, il faut donc agir directement au niveau des bases de données `.bib` et des fichiers de style `.bst`.

**Fichiers `.bib`** : s'assurer que chaque référence des bases de données `.bib` utilisées comporte un champ « `language = {...}` » définissant la langue d'origine de la référence.

**Fichiers `.bst`** : pour remplacer les styles standard `alpha`, `plain`, `unsrt`, faire appel à l'extension `babelbib` et aux styles `babalpha`, `babplain`, `babunsrt` (voir la documentation `babelbib.pdf` et le fichier d'exemples `babelbibtest.tex`). Selon les options, il est possible

d'afficher chaque référence dans sa langue ou bien de les afficher toutes dans la langue principale du document<sup>22</sup>.

Ceux qui font appel à un style de bibliographie sur mesure créé à partir de `custom-bib` devront choisir l'option `babel` lors de la création du fichier `.bst` et ensuite adapter le fichier `babelbst.tex` aux langues utilisées.

## 7.2 Bibliographie avec Biblatex/Biber

La solution Biblatex/Biber présente des nombreux avantages par rapport à `BIBTEX` :

- Biblatex prend en compte les options de Babel, les « *and* » des listes d'auteurs sont transcrits automatiquement en « *et* » dans un document en français ;

- Biblatex, associé à Biber, permet le traitement des fichiers `.bib` codés en utf-8 ce qui facilite grandement la coexistence de références à des ouvrages en français, en russe et en grec par exemple ;

- Biblatex possède des options qui remplacent de nombreuses extensions spécifiques telles que `bibtopic`, `bibunits`, `chapterbib`, `cite`, `multibib`, `natbib`, etc.

Pour la mise en œuvre pratique de Biblatex/Biber, consulter le manuel *LaTeX, l'essentiel* de D. BITOUZÉ et J.-C. CHARPENTIER ou la documentation en anglais `biblatex.pdf`.

## 8 Compatibilité avec e-french

Il est souhaitable qu'un texte saisi avec `e-french` de Bernard GAULLE puisse être compilé avec un minimum de modifications sur une machine utilisant `babel-french` et réciproquement.

En ce qui concerne les guillemets français, `e-french` rend actifs les caractères `<` et `>` afin de saisir les guillemets sous la forme `<<` et `>>` tandis que `babel-french` propose d'utiliser les vrais guillemets chevrons « et » ou la commande `\frquote{}`.

Les points de suspensions sont saisis `. . .` avec `e-french` et `...` (ou `\dots`) avec `babel-french`.

Les commandes suivantes peuvent être ajoutées au préambule pour émuler certaines commandes de `e-french` :

```
\let\numero=\no
\let\Numero=\No
\let\fsc=\bsc
\let\lsc=\bsc
\newcommand*\french{\leavevmode\selectlanguage{french}}
\newcommand*\english{\leavevmode\selectlanguage{english}}
\newcommand*\AllTeX{%
  (L\kern-.36em\raise.3ex\hbox{\sc a}\kern-.15em)%
  T\kern-.1667em\lower.7ex\hbox{E}\kern-.125emX}
```

---

22. Les styles francisés `*-fr.bst` qu'on trouve sur CTAN dans `tex-archive/biblio/bibtex/contrib/bib-fr` n'offrent que la seconde possibilité (références toutes en français).

Pour ceux qui éditent leurs sources LaTeX avec emacs, une fonction Lisp `french2b` opère une adaptation *partielle* d'un fichier conçu pour `e-french` facilitant sa compilation avec `babel-french`. L'appel à `e-french` est remplacé en un appel à `babel-french`, les guillemets `<<` et `>>` sont convertis en `\og` et `\fg` et les `. . .` en `...`, enfin quelques commandes spécifiques à `e-french` sont ajoutées dans le préambule.

Cette fonction est disponible sur <http://daniel.flipo.free.fr/frenchb/french2b.el>. Il suffit de l'ajouter à un fichier `.emacs` et de l'exécuter par `Esc x french2b` sur le fichier à convertir.

Daniel FLIPO  
<http://daniel.flipo.free.fr>