Mode d'emploi du module babel-french

Daniel Flipo

Version 3.7g – 8 novembre 2026

Cette documentation concerne pdf(La)TeX et Xe(La)TeX.

Les utilisateurs de Lua(La)TeX sont invités à consulter le fichier frenchb-fr.pdf.

Résumé

La première version de frenchb (**french** pour **b**abel) 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, 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 et adaption à l'évolution du noyau LaTeX –, l'autre « moderne » frenchb.dtx destinée à Lua(La)TeX, qui continuera à évoluer.

La liste détaillée des changements introduits dans la version 3.7 se trouve à la section 4.1 p. 17.

Historique des mises à jour de cette documentation

22 mai 2024 Adaptation à la version 3.6a, voir section 4.2.

25 mai 2025 Adaptation à la version 3.7a, voir section 4.1.

6 juin 2026 Adaptation à la version 3.7f, voir section 4.1.

Sommaire

1 Appel de l'extension Babel **3**

2 Description de la francisation par babel-french **4**

3 Personnalisation **9**

3.1 \frenchsetup 9

3.2 Traduction des intitulés 14

3.3 Présentation des listes 15

4 Changements entre les versions 3.7g et 3.3 (2018) **17**

4.1 Changements en version 3.7 17

4.2 Changements en version 3.6 18

4.3 Changements en version 3.5 19

4.4 Changements en version 3.4 20

4.5 Changements en version 3.3 20

4.6 Comment recompiler un document écrit pour babel-french 2.x? 21

5 Problèmes de césures **21**

6 Problèmes avec les quatre caractères actifs (; :!?) **21**

7 Incompatibilités connues et remèdes **22**

8 Bibliographie **23**

8.1 Bibliographie avec BibTeX 23

8.2 Bibliographie avec Biblatex/Biber 24

9 Compatibilité avec e-french **24**

1 Appel de l'extension Babel

Babel est installé en standard dans toutes distributions LaTeX, pour disposer des langues française et anglaise[1], il suffit d'ajouter `\usepackage[english,french]{babel}`[2] 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*[3] 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}`[4].

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…*`}`.

Depuis la version 3.10 de Babel, une syntaxe allégée est proposée pour les changements de langue : en ajoutant par exemple dans le préambule `\babeltags{fr = french}`[5], 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}`.

Remarque importante sur les fontes. Le recours à `babel-french` ne suffit pas pour obtenir des césures correctes des mots accentués, il faut en outre utiliser des fontes contenant tous les caractères spéciaux du français (â, é, ï, ù, ç, etc.) ; la façon de le faire dépend du format utilisé :

— avec pdfLaTeX il convient d'ajouter dans le préambule les deux commandes `\usepackage[T1]{fontenc}`

1. En fait américaine (US-english), il existe une variante `british` pour l'anglais « britannique ».
2. Les options `frenchb` et `francais` et `acadian` (équivalentes à `french` depuis 2004), sont définitivement supprimées, la seule option pour le français est `french`.
3. Noter que la présence d'une commande `\DocumentMetadata{lang=..}` impose sa valeur comme langue principale de Babel.
4. Là encore, le nom de la langue française est `french`, pas `frenchb` ou `français`.
5. Rien n'empêche de remplacer `fr` par `french`, on retrouve ainsi la syntaxe de `polyglossia`.

`\usepackage{lmodern}`
rappelons que la déclaration du codage d'entrée n'est nécessaire que si celui-ci n'est pas le codage par défaut (`utf8` depuis 2018), par exemple `latin9` ou `latin1` ou `applemac`, etc.
`\usepackage[latin9]{inputenc}`

— avec XeLaTeX on peut ajouter `\usepackage{fontspec}` mais ceci est devenu facultatif avec les formats LaTeX récents (2017).

Dans les deux cas, le document sera composé avec les fontes LM ou *Latin Modern* qui sont la version moderne des fontes CM ou *Computer Modern*, fontes historiques de TeX. Les fontes LM conviennent pour toutes les langues de l'Europe de l'ouest (latines, anglo-saxonnes et scandinaves).
Rappelons qu'avec XeLaTeX aucun appel à `\usepackage[...]{inputenc}` n'est à faire, le texte source *devant être codé* en `utf8`.

L'utilisation de fontes de la famille CM/LM n'est pas du tout indispensable,

— pour pdfLaTeX toute fonte PostScript en codage T1 convient, ainsi on pourra remplacer `lmodern`, au choix par `kpfonts`, `fourier` (Utopia), `mathptmx` ou `txfonts` (Times), `mathpazo` ou `pxfonts` (Palatino), etc.

— pour XeLaTeX, 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 :

```
\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.
Il est également possible d'utiliser des fontes PostScript, pour plus de détails consulter par exemple le chapitre 15 du livre *LaTeX, l'essentiel* de D. Bitouzé et J.-C. Charpentier.

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.

Depuis la version 2.2, `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[6] :

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

— 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. Depuis la version 2.6a, 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[7] au lieu de ceci[8]. Noter, outre la présentation différente du numéro dans la note elle-même, l'ajout de l'espace fine 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 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).
— Les légendes des figures et des tables utilisent un tiret double à la place du « : », ceci pour toutes les langues, on obtient « Figure 1 – Légende » au lieu de « Figure 1 : Légende ». Il est possible de choisir un autre séparateur : pour remplacer le tiret double « – » par un tiret triple « — », ajouter dans le préambule
`\renewcommand*{\CaptionSeparator}{\space\textemdash\space}`.
Lorsque la langue principale n'est pas le français, le séparateur « : » est utilisé pour toutes les langues mais une espace insécable est en principe ajoutée en français ; lorsque l'ajout échoue, un message est affiché dans le fichier `.log`.
Certaines classes ou extensions modifient la présentation des légendes ; parmi elles, les classes `memoir`, `beamer`, koma-script et AMS et les extensions `caption` et `floatrow` sont compatibles avec `babel-french`.
Un avertissement est inséré dans le fichier `.log` lorsqu'il y a risque de conflit avec une autre extension ou classe qui modifie la définition de la commande `\@makecaption` ou lorsque le chargement d'une extension intervient trop tôt ou trop tard.

En ce qui concerne les conventions typographiques locales (variables avec 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 (`; : ! ?`)[9]. Ceci impose que ces caractères soient rendus actifs (en fait dans tout le document), *sauf* avec les versions récentes de Xe(La)TeX qui disposent d'un

7. Une note de bas de page « à la française ».
[8]Une note de bas de page standard (classe `article`), ça jure avec la précédente, non ?
9. Voir section 3 p. 9 comment éviter les ajouts abusifs d'espace avant ces caractères et section 6 p. 21 en cas de problème dû aux caractères actifs.

mécanisme permettant d'accéder aux mêmes fonctionnalités sans recours aux caractères actifs.

— 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. 14 comment modifier ces intitulés.

— La définition de `\dots` est modifiée[10], *uniquement* avec les moteurs 8-bits (pdftex), afin de supprimer l'espace qui suit, espace gênant devant une parenthèse par exemple. Ceci dit, la commande `\dots` *ne devrait plus être utilisée* dès lors que le texte source est codé en utf-8 : il est plus facile de saisir … [11] que `\dots`, le texte source est plus lisible et on évite les problèmes d'avalement des espaces après une commande. Avec le moteur XeTeX, la commande `\dots` standard utilise le glyphe de la police de sortie sans ajout d'espace, `babel-french` ne la redéfinit donc pas.

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[12], ils peuvent être utilisés pour saisir les guillemets français, voir la fin de la section 3, p. 12.
Notez que les guillemets français ne devraient *jamais* être saisis avec << et >> contrairement à ce qui se fait avec e-French (voir section 9).
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).

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{}`.

10. en mode texte seulement, en fait c'est `\textellipsis` qui est modifiée.
11. Il s'agit du glyphe … obtenu par exemple grâce à une touche `Compose`, *pas* de . . . (trois points consécutifs).
12. grâce à une touche `Compose` par exemple…

Exemple de citation imbriquée :

> 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 *surbooking* en anglais. › »

`InnerGuillSingle`

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^{me}, `1\up{er}` donne 1er ; on dispose aussi de `\ier \iere \iers \ieres \ieme \iemes` pour 1er, 1re, 1ers, 1res, 2^{e}, 2es. La commande `\up` utilise les lettres supérieures de la police lorsqu'elles sont disponibles et les simule sinon. On obtient de vraies lettres supérieures

— avec le moteur XeTeX, en chargeant les extensions `fontspec` *et* `realscripts` et en choisissant une police de type OpenType qui connaît 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``) ;

— sous (pdf)LaTeX avec certaines polices type 1 expertes (Fourier-Utopia).

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 ou la police type 1 Utopia-expert par exemple, n'ont pas de « g supérieur » ; en codant `M\up{gr}` on obtient M^{gr} (Jenson) ou M^{r} (Utopia-expert) tandis que `M\up*{gr}` force l'utilisation de supérieures simulées ce qui pallie l'absence du « g supérieur », le résultat est M^{gr}.

Pour les perfectionnistes, il est facile d'ajuster la taille et le placement vertical des lettres supérieures simulées par `\fup` (voir dans le code de `frenchb.dtx` les commandes `\FBsupS` et `\FBsupR`).

— 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° , 2° , 3° , 4° . Ensuite, `\FrenchEnumerate{6}` donne 6° .

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

— Le caractère ° (à ne pas confondre avec le petit « o » de N°) est saisi `\degre`, mais

comme les espaces avant et après ce caractère dépendent fortement de la police utilisée (PostScript ou non), on emploiera `\degres` pour saisir les températures « 20 °C » (codé `20~\degres C` avec espace insécable) ou les titres alcooliques « 45° » (codé `45\degres` sans espace). Lorsque l'extension `textcomp` est chargée (elle donne accès aux « TEX Companion fonts » qui contiennent un vrai symbole degré), `\degres` utilise celui-ci (`\textdegree`).

— La commande `\frenchdate{`*année*`}{`*mois*`}{`*jour*`}`, qui prend trois arguments numériques obligatoires, affiche les dates en français : `\frenchdate{2001}{01}{01}` donne 1er 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[13]. 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`,…, `\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*. Un exemple extrême est fourni par la commande `\dots` : lorsqu'elle est redéfinie par `babel-french` (moteurs 8-bits et en français seulement) on peut omettre la paire d'accolades, mais celle-ci est indispensable dans une partie en anglais par exemple ou avec les moteurs Unicode (`\dots` garde sa définition standard)… Il me paraît plus simple d'éviter le recours à `xspace` et de gérer les espaces soi-même : « `les 1\ier, 2 et 3~mai` » ou « `le 1\ier~mai` » (espace insécable).

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

3 Personnalisation

La commande \frenchsetup{}, appelée précédemment \frenchbsetup{}[14], 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 a été supprimé en version 3.0.

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 (dans le cas où le français est la langue principale) toute action de babel-french sur la maquette : retour aux listes standard, pas de retrait des 1^ers paragraphes des sections, notes de bas de page standard, séparateur « : » dans les légendes de figures et tableaux. 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.

14. 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.

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. 16 pour une comparaison avec la disposition par défaut.

StandardListSpacing=true (false*) ; par défaut babel-french modifie (en général en les réduisant, sauf pour les classes SMF) les espaces verticaux[15] dans *toutes les listes* produites à partir de l'environnement list, en particulier les listes itemize, enumerate 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[16]). 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 16.

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[17]).

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, \ding{43},... (\textemdash*)

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

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

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

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

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

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

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 quatre sous-options plus ciblées qu'elle regroupe) devrait régler le problème [18].

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 affecte la totalité du document. La commande \StandardFootnotes peut encore être utilisée *localement*, par exemple dans les environnements minipage si la présentation « à la française » ne convient pas (notes numérotées 'a', 'b', etc.).

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) ; par défaut, babel-french corrige la faute de saisie qui consiste à omettre l'espace devant la ponctuation haute (:;?!) en ajoutant automatiquement une espace insécable de taille adaptée. Toutefois, l'ajout d'espace n'a pas lieu lorsque la police courante est à espacement fixe [19] : \texttt{http://truc} produit `http://truc`.

Ceux qui sont sûrs de leur saisie peuvent mettre cette option à false pour contrôler complètement l'ajout d'espace devant la ponctuation haute : l'espace adéquate (toujours insécable) est ajoutée si et seulement si un espace précède le signe de ponctuation dans le fichier source. Ils éviteront ainsi, quel que soit le moteur utilisé, de voir des espaces ajoutées à tort.

Autre solution pour les autres : laisser le mode par défaut et éviter, le cas échéant, l'insertion d'espaces parasites en utilisant la commande \NoAutoSpacing dans un groupe, par exemple {\NoAutoSpacing 10:55}.

ThinColonSpace=true (false) ; par défaut l'espace placée avant le ':' 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 le ':'.

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 un argument optionnel (pour le dialecte french ou acadian, son absence équivaut à french) et quatre arguments obligatoires : le premier précise le type d'espace à redéfinir, il doit être colon (pour le ':'), 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 ';'

18. L'option StandardLists est automatiquement activée avec la classe beamer.
19. Ceci évite en particulier les problèmes en verbatim.

‘!’ 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).

`\FBsetspaces{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 ‘« ’, ni avant ‘ »’ même si ces caractères ont été activés par l’option `og=«, fg=»`. De plus, les espaces présents dans 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, ce qui peut être utile pour recompiler à l’identique des anciens textes.

Guillemets

`og=«, fg=»`[20] ; lorsqu’on dispose de guillemets français au clavier (grâce à une touche *compose* par exemple), cette option permet d’utiliser directement ces guillemets à la place des commandes `\frquote{}` ou `\og` et `\fg`. On peut ainsi saisir `«guillemets»` (sans espace) ou `« guillemets »` (avec espaces à l’intérieur) pour obtenir « guillemets » avec les espaces insécables adéquates en français. En revanche, si on active cette option, *il ne faut pas coder explicitement les espaces insécables* coder `«~guillemets~»` produirait une espace trop large après le guillemet ouvrant.

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

Ceci fonctionne en (pdf)LaTeX pour les codages d’entrée 8-bits (latin1, latin9, ansinew, applemac,…) et pour les codages sur plusieurs octets comme utf8 ou utf8x. Ceci fonctionne également avec XeLaTeX ; avec XeLaTeX toutefois, comme pour la ponctuation haute, `babel-french` respecte les espaces insécables U+00A0 et U+202F (fine).

`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 `\FBsetspaces{guill}{1}{1}{1}`, voir p. 11.

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

`InnerGuillSingle=true (false)` ; cette option modifie les guillemets utilisés pour les citations internes : comme ‹ ceci › (`true`), ou comme “cela” par défaut (`false`).

`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.

20. Les valeurs affectées à `og` et `fg` sont les vrais guillemets « et », pas << et >>. Les espaces avant et après ces guillemets dans la commande `\frenchsetup{}` sont optionnels.

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=false (true*) ; si cette option est mise à `false`, le séparateur défini par `\CaptionSeparator` est remplacé par le séparateur par défaut (deux-points) dans les légendes des figures et des tables, ceci pour toutes les langues. En français, `babel-french` ajoute si possible une espace insécable adéquate devant le deux-points ou affiche un message dans le fichier `.log`.

OldFigTabCaptions=true (false) peut être utilisée pour retrouver la présentation antérieure (pré 3.0) des légendes de figures et tables, c.-à-d. `\CaptionSeparator` en français et deux-points pour les autres langues. Cette option ne fonctionne que pour les classes standard `article`, `report`, `book` et rend inopérante l'option `CustomiseFigTabCaptions`.

Divers : lettres supérieures et « warnings »

FrenchSuperscripts=false (true) ne devrait être utilisée que pour recompiler des anciens fichiers à l'identique. Elle redéfinit `\up` comme `\textsuperscript` alors que par défaut `\up` fait appel à la commande `\fup` plus conforme aux usages francophones (voir section 2, p. 4.

LowercaseSuperscripts=false (true) rend possible d'avoir des lettres supérieures en capitales (est-ce bien utile ?). Par défaut, la commande `\up` (sauf si elle est redéfinie en `\textsuperscript` par l'option précédente) 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 renfoncement au début des 1ers 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,AutoSpaceFootnotes=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
\abstractname	\frenchabstractname	Résumé
\bibname	\frenchbibname	Bibliographie
\refname	\frenchrefname	Références
\prefacename	\frenchprefacename	Préface
\chaptername	\frenchchaptername	Chapitre
\appendixname	\frenchappendixname	Annexe
\contentsname	\frenchcontentsname	Table des matières
\listfigurename	\frenchlistfigurename	Table des figures
\listtablename	\frenchlisttablename	Liste des tableaux
\indexname	\frenchindexname	Index
\glossaryname	\frenchglossaryname	Glossaire
\figurename	\frenchfigurename	Figure
\tablename	\frenchtablename	Table
\partname	\frenchpartname	Première partie, Deuxième partie…
\enclname	\frenchenclname	P. J.
\ccname	\frenchccname	Copie à
\headtoname	\frenchheadtoname	< *vide* >
\pagename	\frenchpagename	page
\seename	\frenchseename	voir
\alsoname	\frenchalsoname	voir aussi
\proofname	\frenchproofname	Démonstration

Il est facile de modifier ces traductions : pour remplacer « Démonstration » par « Preuve » (avec amsthm) ajouter dans le préambule \def\frenchproofname{Preuve}[21], noter que l'ancienne syntaxe \addto\captionsfrench{\def\proofname{Preuve}} fonctionne toujours.

Attention, dans les deux cas, le nom de la langue est *obligatoirement* french, les variantes frenchb ou francais seraient sans effet !

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}[21] ; 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}[21]. Le remplacement de « Deuxième » par « Seconde » est possible en codant \def\frenchpartsecond{Seconde}[21] dans le préambule. De même il est possible de remplacer « Première » par « Premier » en codant \def\frenchpartfirst{Premier}[21].

On remarquera que \figurename et \tablename sont en petites capitales en français. Il serait préférable, notamment dans un document multilingue, de s'en tenir à la traduction pure (\def\frenchfigurename{Figure}[21], \def\frenchtablename{Table}[21] et de choisir les attributs de la police (\scshape par exemple) au niveau de la classe ou d'une extension

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

comme captions. L'option SmallCapsFigTabCaptions (p. 13) peut être mise à false pour supprimer le passage en petites capitales.

3.3 Présentation des listes

Voici la présentation par défaut[22] pour les listes itemize (tiret cadratin et alignement des tirets de premier niveau sur le retrait (\parindent) de début de paragraphe) :

<= 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, \textemdash par défaut, \textendash, ou autre.

Deux 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

\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}.

22. L'option ListItemsAsPar permet une autre présentation, voir ci-dessous p. 16.

<= 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 placées 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 deux lignes…
— 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}{\textemdash}  % 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 3.7g et 3.3 (2018)

4.1 Changements en version 3.7

`frenchb` est maintenant scindée en deux parties, une partie « historique » réservée aux moteurs anciens (TeX, pdfTeX et XeTeX) qui restera figée (`frenchb3.dtx`) — sauf corrections éventuelles de bogues et adaption à l'évolution du noyau LaTeX –, l'autre « moderne » (`frenchb.dtx`) destinée à Lua(La)TeX qui continuera à évoluer. Cette scission a permis de simplifier le code pour LuaTeX et de le débarrasser de vieilleries devenues inutiles mais également d'alléger `frenchb3.dtx` en supprimant le support pour LuaTeX.

Le support de la variante `acadian`[23], est abandonné ainsi que les alias `frenchb` et `francais`. La seule option acceptée désormais pour le français est `french`.

Les options `ListOldLayout`, `CompactItemize` et `GlobalLayoutFrench` ont été supprimées car elles ne servaient qu'à reproduire des comportements très anciens de `babel-french`.

Note sur le projet « PDF tagging[24]**» :** ce projet de l'équipe LaTeX3 nécessite une réécriture complète de nombreuses parties du code LaTeX. Le nouveau code est construit à partir de gabarits (*templates*) qui se déclinent en instances (*instances*) *flexibles*.

Le vœu de l'équipe LaTeX3 est de proposer suffisamment de flexibilité pour que les auteurs de classes ou d'extensions n'aient plus à *redéfinir* les commandes ou environnements de base mais seulement à *modifier* leurs paramètres de configuration. Ainsi les balises ajoutées automatiquement dans le fichier PDF seront préservées et les incompatibilités entre redéfinitions concurrentes seront évitées.

23. Et de son synomyme `canadien`

24. Il s'agit de produire des fichiers PDF conformes aux standards d'archivage (A-4) et d'accessiblité (UA-2) et dotés de balises de structure.

En version 3.7a la personnalisation des listes a été désactivée et un message l'annonçait dans le fichier `.log` dès que les nouvelles listes étaient utilisées, c.-à-d. en présence de la ligne `\DocumentMetadata{testphase=latest}` ou `\DocumentMetadata{tagging=on}`

La version 3.7b 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 (composé avec LuaLaTeX) contient

```
\DocumentMetadata{pdfstandard={A-4,UA-2}, tagging=on, lang=fr}
\documentclass[a4paper,12pt,german,british,french]{article}
\usepackage{babel}
```

Le changement majeur dans la version 3.7f est l'ajout de code permettant de paramétrer les listes à partir de la nouvelle interface pour obtenir un rendu analogue à ce qui est produit par les versions précédentes de `babel-french`. Il faut savoir que cette nouvelle interface a vocation à remplacer, dans un futur plus ou moins proche, les anciennes structures au sein du noyau LaTeX. Actuellement, ce nouveau code n'est pris en compte que si le document commence par une commande `\DocumentMetadata{...}`, rien ne change donc ni pour les utilisateurs de XeLaTeX[25], ni pour ceux de pdfLaTeX qui ne sont pas intéressés par le balisage des fichiers PDF.

Ceux qui souhaitent expérimenter à ce sujet devraient plutôt opter pour LuaLaTeX et consulter la documentation française `frenchb-fr.pdf` de la version 4.1a de `babel-french` dédiée au moteur LuaTeX. Ceci dit, rien n'empêche de tester la nouvelle interface avec pdfLaTeX en ajoutant par exemple une commande `\DocumentMetadata{...}` avant la commande `\documentclass{}`, mais il faut alors utiliser une distribution très récente car le format LaTeX 2026-06-01 est indispensable.

4.2 Changements 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`[26] 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 $(1/6 - 1/8) \cdot 12 = 0{,}5$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}{\,}`.

25. Le balisage des fichiers PDF n'est pas possible avec le moteur XeTeX qui ne produit pas directement de fichiers PDF.
26. Elles sont définies pour fonctionner aussi en mode mathématique.

4.3 Changements en version 3.5

La version 3.5a propose une nouvelle option `ListItemsAsPar` qui permet une présentation des listes plus conforme à la tradition typographique française, voir p. 16. La présentation par défaut des listes est inchangée.

Quelques bogues affectant la commande `\frquote{}` ont été corrigées dans les versions 3.5b à 3.5d. Cette dernière introduit une nouvelle option `StandardListSpacing` à utiliser à la place de `ReduceListSpacing`, voir p. 10. La présente documentation est maintenant incluse dans la distribution `babel-french` sur CTAN.

La commande `\NoEveryParQuote` a été ajoutée en version 3.5e, voir p. 6.

La version 3.5g corrige une bogue ancienne affectant l'usage des polices type 1 avec Lua(La)TeX : tout crénage était supprimé pour ces polices depuis la version 3.1f (2015) ; les polices OpenType elles, n'ont jamais été affectées.

La version 3.5j corrige aussi une bogue ancienne affectant les classes koma-script, `memoir` et `beamer` : les redéfinitions du séparateur des légendes de figures et tables (commandes `\captionformat`, `\captiondelim`, etc.) sont maintenant prises en compte correctement.

À partir de la version 3.5k :

— La traduction française de `\figurename` et `\tablename` ne contient plus de commande de changement de fonte comme c'est la règle. Le passage en petites capitales a été déplacé dans `\fnum@figure` et `\fnum@table`, ce qui a pour effet de mettre le numéro également en petites capitales. Noter que la classe `beamer` ne reconnaît pas les commandes `\fnum@...`, les légendes ne sont donc plus en petites capitales.

— Le chargement de `caption` peut se faire indifféremment avant ou après celui de `babel`.

— La commande `\pdfstringdefDisableCommands` n'est plus utilisée : toutes les commandes nécessitant un traitement spécial dans les signets `hyperref` sont maintenant définies à partir de `\textorpdfstring{}{}` (suggestion de l'équipe LaTeX3).

La commande `\bname{}` (voir section 7) a été ajoutée en version 3.5n, la documentation concernant la commande `\bsc{}` a été revue comme celle concernant la saisie des guillemets.

À partir de la version 3.5o :

— Le codage (déconseillé) «~abc~» ne produit plus d'espace parasite *avec Lua(La)TeX*, ceci répond à la demande formulée ici.

— Les commandes `\shorthandoff{}` et `\shorthandon{}` affichent maintenant le message standard de `babel` lorsqu'elles sont exécutées (à tort) sous LuaTeX ou XeTeX. Leur redéfinition, qui permettait de conseiller `\NoAutoSpacing` à la place, a été supprimée à la demande de Javier Bezos, car elle cassait la variante `\shorthandoff*{}`.

Le comportement de `\DecimalMathComma` a changé en version 3.5p suite à une remarque de Fabrice Eudes : elle peut désormais être utilisée dans le préambule. Voir section 2 p. 8 pour plus de détails.

La version 3.5q corrige un bug concernant les listes signalé par Denis Bitouzé : les alinéas inclus dans une liste n'étaient pas distinguables. Dorénavant `\listparindent` est défini comme `\parindent` et si`\parskip > 0`, `\parsep` est défini comme `\parskip`. Il est possible

de revenir à l'ancienne présentation en ajoutant *dans l'environnement de liste* les commandes `\parskip=0pt` et `\parindent=0pt`.

La version 3.5r est compatible avec `ucharclasses.sty` qui est maintenant chargé avec l'extension `fontsetup` lorsque le moteur XeTeX est utilisé. D'autre part le fichier `frenchb.ins` n'est plus nécessaire pour extraire les fichiers `*.ldf` de `frenchb.dtx` (voir `README.md`).

La version 3.5s désactive l'option `FrenchFootnotes` lorsque l'extension `footnotebackref` est chargée, les deux étant incompatibles ; un message d'avertissement est affiché dans le fichier `.log`.

4.4 Changements en version 3.4

La version 3.4a introduit une nouvelle commande `\frenchdate` (voir p. 8) et corrige une bogue mineure : `\FBthousandsep` n'est plus une espace justifiante insécable mais une espace rigide (de même valeur). Le retour à l'ancien comportement est obtenu par l'ajout dans le préambule de `\renewcommand*{\FBthousandsep}{~}`.

Les deux options `french` et `acadian` peuvent maintenant être utilisées simultanément dans un même document ; elles ne présentent par défaut aucune différence mais rien n'empêche de les personnaliser différemment en ce qui concerne les motifs de césures, les légendes ou la ponctuation.

Une commande `\FBsetspaces` a été ajoutée pour faciliter le réglage des espaces précédant la ponctuation haute et les guillemets avec la possibilité de réglages différents selon l'option `french` ou `acadian`, voir p. 11.

À partir de La version 3.4, eTeX et LuaTeX 1.0 sont requis.

4.5 Changements en version 3.3

Le contrôle de l'ajout d'espace devant le deux-points a été amélioré dans la version 3.3d mais uniquement avec le moteur LuaTeX : il n'y a plus d'ajout d'espace parasite dans les URL (http://monsite), les chemins MSDOS (C:\Mes Documents) ou les horaires (10:55).

Lors de la conversion en HTML par `lwarp` (versions 0.37 et suivantes) des fichiers compilés sous XeLaTeX ou pdfLaTeX, les espaces insécables ajoutées par `babel-french` pour la ponctuation haute et les guillemets sont respectées. Une nouvelle option (expérimentale) `UnicodeNoBreakSpaces` pour LuaLaTeX a été ajoutée dans la version 3.3c.

La version 3.3b a subi un réaménagement interne : chaque langue ou dialecte de Babel devrait avoir son propre fichier `.ldf`. Ainsi pour le français il y a dorénavant `french.ldf` qui contient le code principal et quatre fichiers satellites `frenchb.ldf`, `francais.ldf`, `acadian.ldf` et `canadien.ldf`. L'option à utiliser pour charger le français avec Babel est toujours `french` (ou `acadian` pour le français du Canada pour l'instant identique au français de base), les autres options (`frenchb`, `francais` ou `canadien`) sont déconseillées et affichent un message d'avertissement dans le fichier `.log`.

La version 3.3a disponible dans TeXLive 2017 est compatible avec LuaTeX 1.0.4 (stable) mais aussi avec la version 0.95 (beta) de TL2016. L'espacement de la ponctuation haute

et des guillemets français est maintenant contrôlé par les *commandes* `\FBcolonspace`, `\FBthinspace` et `\FBguillspace` quel que soit le moteur utilisé (LuaTeX, XeTeX ou pdfTeX) ; auparavant LuaTeX avait recours à des *glue* `\FBcolonskip`, `\FBthinskip` et `\FBguillskip`.

L'option `french` devant être préférée à `frenchb`, il m'a semblé judicieux de renommer la commande de personnalisation `\frenchbsetup{}` en `\frenchsetup{}`, l'ancien nom étant conservé par souci de compatibilité.

De nouvelles possibilités de personnalisation de la commande `\part{}` ont été introduites, voir page 14.

4.6 Comment recompiler un document écrit pour babel-french 2.x ?

Penser d'abord à remplacer `frenchb` et `francais` par `french` dans les options de Babel et dans `\selectlanguage{...}`, `\begin{otherlanguage}{...}`, `\foreignlanguage{...}`. Une exception toutefois : les classes SMF (`smfart` et `smfbook`) ne fonctionnent qu'avec l'option `frenchb`… vieux reliquat d'un passé lointain où l'option `french` de babel était ambiguë (ambiguïté levée depuis 2004).

Ajouter `\listindentFB=0pt` juste avant les environnements `description` si on souhaite que les étiquettes de ces environnements collent à la marge gauche au premier niveau.

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 `https://danielflipo.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 Problèmes avec les quatre caractères actifs (; :!?)

Rappelons d'abord que ces quatre caractères *ne sont pas rendus actifs* avec le moteur XeTeX, donc aucun problème n'est à craindre avec celui-ci[27], cette section ne concerne *que les vieux moteurs* TeX'82 et pdfTeX.

27. N'est-ce pas une bonne raison d'abandonner pdf(La)TeX ? Passer à LuaTeX me semble s'imposer…

Normalement, le nécessaire est fait par Babel pour que les caractères rendus actifs ne perturbent pas les autres extensions… mais il y a hélas des exceptions (tikz, xypic, xcolor, autonum, arabtex, cleveref par exemple). Il faut savoir que les caractères rendus actifs par une langue *le restent dans tout le document*, repasser en anglais par exemple ne désactive pas les ; :!? s'ils ont été rendus actifs par babel-french !

Les espaces insécables ajoutés entre les guillemets peuvent également être gênants.

Depuis la version 2.5, la commande à utiliser en cas de problème est \NoAutoSpacing[28] *dans un groupe* comme ceci :

```
{\NoAutoSpacing
Partie ne supportant pas les caractères actifs
}
```

ou à l'intérieur d'un environnement, par exemple

```
\begin{tikzpicture}\NoAutoSpacing
 ...
\end{tikzpicture}
```

Avec tikz v3.0, une solution alternative consiste à ajouter \usetikzlibrary{babel} dans le préambule, voir la documentation de TikZ (pgfmanual.pdf). L'avantage de cette solution est de tenter de préserver les caractères actifs dans les nœuds même si ça ne fonctionne pas toujours, notamment pour le point d'exclamation.

Avec pdf(La)TeX, il est également possible de désactiver un seul ou plusieurs caractères actifs de manière sélective grâce à la commande \shorthandoff de Babel. Le mieux est de le faire *localement* dans un environnement ou dans un groupe comme ceci :

```
{\shorthandoff{:!}%
Partie ne supportant pas les caractères actifs ':' et '!'
}
```

7 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.

— Les caractères rendus actifs par babel-french[29] (; :!?) peuvent perturber certaines extensions, c'est le cas de tikz, xypic, xcolor, arabtex, cleveref par exemple, voir comment y remédier section 6.
En règle générale, il vaudrait mieux éviter les caractères actifs dans les \label et \caption, remplacer systématiquement les « : » par des tirets « - » dans les \label est une saine précaution. Pour éviter les problèmes de caractères actifs avec natbib ou listings, il convient de charger natbib *avant* babel et listings *après* babel. Un message est affiché dans le fichier .log lorsque l'ordre de chargement n'est pas correct.
hyperref ne fonctionne pas avec la commande \cite standard lorsque celle-ci contient un caractère actif, charger l'extension cite règle le problème.

28. Elle fonctionne avec les formats pdf(La)TeX et et Xe(La)TeX mais ne règle pas tous les problèmes de caractères actifs, notamment dans les \label et autres \caption, voir la section 7 à ce sujet. .
29. Ceci ne concerne pas Xe(La)TeX qui gère autrement la ponctuation haute.

— En LaTeX, les caractères ; :!? ne sont rendus actifs qu'au `\begin{document}`, ainsi les espaces attendues ne sont pas ajoutées automatiquement lorsque ces caractères sont utilisés dans des commandes définies dans le préambule ou dans des fichiers `.sty`.
Exemple : la commande `\title{Quelle crise?}` placée dans le préambule imprimera « Quelle crise? » (sans espace)[30] lors de l'appel de `\maketitle`. Il y a plusieurs parades :

— soit placer la commande `\title{Quelle crise?}` après le `\begin{document}` (et avant `\maketitle`),

— soit placer les commandes du type `\title` entre un `\shorthandon{;:!?}` (avant) et un `\shorthandoff{;:!?}` (après), si on tient à les laisser dans le préambule.

— Les extensions `caption`, `subcaption`, `floatrow` doivent impérativement être chargées *après* `babel`. En revanche `beameraticle` 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 soit une erreur à la compilation (en LaTeX), soit un mauvais classement de l'entrée `«toto»` dans l'index (en XeLaTeX).

8 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.

8.1 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

30. Mais l'espace sera préservée si on compile en en XeLaTeX puisque le mécanisme est différent.

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[31].
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.

8.2 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`.

9 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{}`.

Lorsqu'on travaille en codage `T1` avec `babel-french`, l'existence des ligatures `<<` et `>>` permet de saisir `<<~guillemets français~>>`, mais les espaces insécables sont *indispensables*. Les commandes `\og` et `\fg` (ou leurs alias « et », à condition que ceux-ci aient été activés dans `\frenchsetup{}`) sont préférables : elles introduisent automatiquement des espaces insécables et plus fines pour un meilleur rendu typographique.

31. 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).

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}
```

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 `https://danielflipo.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
https://danielflipo.fr