

# Documentation sur le module frenchb de Babel

Daniel FLIPO

Version 2.5g – 21 février 2012

## Résumé

La première version de frenchb (**french** pour **babel**) est sortie en 1996. Depuis, frenchb a beaucoup évolué, notamment grâce aux suggestions des utilisateurs, jusqu'à la version 1.6h (novembre 2006).

La version 2, profondément remaniée, est sortie en mai 2007. Ce qui était devenu obsolète (par exemple le support pour LaTeX-2.09 ou pour mLaTeX) a été supprimé et les multiples options de personnalisation ajoutées au fil du temps ont été regroupées en une seule commande `\frenchbsetup{}`. La liste détaillée des changements intervenus depuis la version 1.6 se trouve à la section 4 p. 8.

Le fichier `frenchb.ldf` utilisé par Babel est produit à partir de deux fichiers, `frenchb.dtx` et `frenchb.ins` qu'on trouvera sur <http://daniel.flipo.free.fr/frenchb>; le premier contient le code commenté et la documentation complète en anglais, le second permet d'en extraire `frenchb.ldf`.

## 1 Appel de l'extension Babel

Babel est installé en standard dans toutes distributions  $\text{\TeX}2_{\epsilon}$ , 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 également possible de déclarer les options de langues comme arguments de `\documentclass`, elles peuvent alors être utilisées également par d'autres extensions :

```
\documentclass[12pt,english,french]{article}
\usepackage{varioref}
\usepackage{babel}
a le même effet que
\documentclass[12pt]{article}
\usepackage[french]{varioref}
\usepackage[english,french]{babel}
```

La dernière langue chargée (le français dans l'exemple ci-dessus) est active au début du document; c'est la *langue principale*, celle qui régit la présentation générale (listes, notes de bas de page, retrait des premiers paragraphes) quelle que soit la langue courante.

---

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, elles ne devraient plus être utilisées sur les installations modernes.

*Remarque importante* : les fontes utilisées par défaut par LaTeX (appelées CM ou *Computer Modern* et correspondant au codage OT1 à 128 caractères) ne permettent pas d’obtenir des césures correctes des mots contenant des signes diacritiques (accents, cédilles, etc.). Pour travailler en français, le recours à Babel/frenchb ne suffit pas, il faut en plus utiliser des fontes contenant tous les caractères spéciaux du français (â, é, ï, ù, ç, etc.), ce qui peut se faire en ajoutant les deux commandes

```
\usepackage[T1]{fontenc}
```

```
\usepackage{lmodern}
```

dans le préambule du document. Ceci impose l’utilisation des fontes LM ou *Latin Modern*, à 256 caractères, qui conviennent pour toutes les langues de l’Europe de l’ouest (latines, anglo-saxonnes et scandinaves).

L’utilisation de fontes de la famille CM/LM n’est pas du tout indispensable, toute fonte de codage T1 convient, ainsi on pourra remplacer `lmodern`, au choix par `kpfonts`, `fourier` (Utopia), `mathptmx` ou `txfonts` (Times), `mathpazo` ou `pxfonts` (Palatino), etc.

Pour passer d’une langue à l’autre en cours de document on utilise la commande standard de Babel `\selectlanguage{langue}`, par exemple `\selectlanguage{french}` et `\selectlanguage{english}`. 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...}.
```

## 2 Description de la francisation par Babel/frenchb

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 double et d’autres, comme 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, frenchb 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, frenchb 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 globale du document (ou maquette) est modifiée de la façon suivante<sup>3</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 doubles « – », ou par un autre marqueur choisi par l’utilisateur (voir section 3) ;

---

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

- les espaces verticaux ajoutés par LaTeX entre les différents éléments d’une liste (*items*) sont supprimés.
- Par défaut les espacements verticaux de *toutes* les listes standard ou créées par l’utilisateur sont réduits.
- Les notes de bas de page sont présentées « à la française » comme ceci<sup>4</sup> au lieu de ceci<sup>5</sup>. Noter, outre la présentation différente du numéro dans la note elle-même, 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 1.5em, 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).

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 sont ajoutés automatiquement (ou seulement rendus insécables et de taille adéquate) devant la ponctuation haute (; : ! ?). Ceci impose que ces caractères soient rendus actifs (en fait dans tout le document)<sup>6</sup>, *sauf* avec les versions récentes de XeTeX ou XeLaTeX qui disposent d’un mécanisme permettant d’accéder aux mêmes fonctionnalités sans recours aux caractères actifs.
- la définition de `\dots` est modifiée pour supprimer l’espace qui suit (cet espace est gênant devant une parenthèse par exemple...)<sup>7</sup>.
- La commande `\today` donne la date en français.
- Les titres (« caption names » en anglais) sont traduits en français, ainsi par exemple la commande `\chapter` imprimera « Chapitre » au lieu de « Chapter ». Il est possible de modifier les intitulés, par exemple, si on souhaite remplacer « Démonstration » par « Preuve » (avec `amsthm`), il suffit d’ajouter dans le préambule `\addto\captionsfrench{\def\proofname{Preuve}}`.  
C’est bien compliqué, pourquoi pas tout simplement `\def\proofname{Preuve}` ou `\renewcommand*\proofname{Preuve}` ? Il y a deux (bonnes) raisons à cela :
  1. si une telle redéfinition est faite dans le préambule, elle est *sans effet* puisque le basculement en français n’a lieu qu’au `\begin{document}` (pour ne pas perturber le chargement des autres extensions) ;
  2. si elle est faite dans le document dans une partie en français, le changement sera effectif dans cette partie mais ne survivra pas lors d’un retour en français après un passage par une autre langue.

Lorsqu’on utilise la syntaxe proposée, la redéfinition de `\proofname` est faite à chaque fois qu’on commute la langue courante vers le français.

- Les légendes des figures et des tables utilisent un tiret double en français à la place du « : » on obtient « Figure 1 – Légende » au lieu de « Fig 1: Légende ». Il est possible de choisir un autre séparateur : par exemple pour remplacer le tiret double « – » par un tiret triple « — »,

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

<sup>5</sup>Une note de bas de page standard (classe `article`), ça jure avec la précédente, non ?

6. Voir section 3 p. 5 comment éviter les ajouts abusifs d’espace avant ces caractères et section 5 p. 12 en cas de problème dû aux caractères actifs.

7. Certains auteurs préfèrent en français « ... » (codé . . . sans espace) à « ... » (codé `\dots`).

ajouter dans le préambule

```
\addto\captionsfrench{\def\CaptionSeparator{\space\textendash\space}}
```

Les légendes des figures et des tables sont laissées intactes si la classe utilisée ou une extension a modifié la définition de la commande `\@makecaption`.

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

Des commandes ont été prévues pour faciliter la saisie :

- Les guillemets peuvent être saisis grâce aux commandes `\og` et `\fg`<sup>8</sup>, ce qui donne selon la langue de travail « guillemets français » ou “english quotes”. L’extension `csquotes` propose une alternative `\enquote{guillemets}`. Les guillemets français ne devraient *jamais* être saisis avec `<<` et `>>` contrairement à ce qui se fait avec FrenchPro (voir section 9).
- Une commande `\up` facilite la saisie des exposants en mode texte : `M\up{me}` imprime M<sup>me</sup>, `1\up{er}` donne 1<sup>er</sup> ; on dispose aussi de `\ier \iere \iers \ieres \ieme \iemes` pour 1<sup>er</sup>, 1<sup>re</sup>, 1<sup>ers</sup>, 1<sup>res</sup>, 2<sup>e</sup>, 2<sup>es</sup>. La commande `\up` a été modifiée (voir les détails section 4.2).
- L’usage français veut que les noms propres soient imprimés en petites capitales et ne soient pas coupés en bout de ligne, la commande `\bsc` (*boxed small caps*) en facilite la saisie : on saisira par exemple `Leslie~\bsc{Lamport}` pour obtenir Leslie LAMPOR<sup>T</sup> sans risque de coupure LAM-PORT en bout de ligne.
- les commandes `\primo`, `\secundo`, `\tertio` et `\quarto` peuvent être utilisées dans les énumérations ; elles donnent 1<sup>o</sup>, 2<sup>o</sup>, 3<sup>o</sup>, 4<sup>o</sup>. Ensuite, `\FrenchEnumerate{6}` donne 6<sup>o</sup>.
- Les abréviations de « numéro », N<sup>o</sup>, N<sup>os</sup>, n<sup>o</sup> et n<sup>os</sup>, sont obtenues en tapant `\No`, `\Nos`, `\no` et `\nos` ; l’espace qui suit est insécable : coder `\no1` ou `\no 1`.
- Le caractère ° (à ne pas confondre avec le petit « o » de N<sup>o</sup>) 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.sty` est chargée (elle donne accès aux « TeX Companion fonts » qui contiennent un vrai symbole degré), `\degres` utilise celui-ci (`\textdegree`).
- En mode mathématique, la virgule est toujours suivie d’un espace car elle est traitée comme un signe de ponctuation et non comme une virgule décimale. La commande `\DecimalMathComma` supprime cet espace tandis que `\StandardMathComma` permet de revenir au comportement par défaut. Remarque qu’après une commande `\DecimalMathComma`, il est nécessaire de saisir un espace explicite dans les listes et les intervalles par exemple `$(x, \ y)$` et `$$[0, \ 1]$$`. Ces deux commandes sont utilisables n’importe où dans le document.
- 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 frenchb à cause du risque de conflit d’options. Les utilisateurs devraient s’habituer progressivement à utiliser `\numprint` (ou son raccourci `\np`) à la place de `\nombre`.

Il est recommandé de profiter des avantages offerts par l’extension `xspace` (il suffit d’ajouter `\usepackage{xspace}` dans le préambule) : les espaces suivant les commandes `\ier`, ..., `\ieres`, `\ieme`, `\iemes`, `\fg` et `\dots` seront respectés sans avoir à les forcer par des `{}` ou des `\_`.

---

8. Si les caractères « et » sont accessibles au clavier, ils peuvent être utilisés pour saisir les guillemets, voir la section 3.

### 3 Personnalisation

Comme il est d'usage avec Babel, le comportement de frenchb peut être personnalisé grâce à un fichier de configuration `frenchb.cfg` (vide par défaut).

Cette possibilité, conservée par souci de compatibilité avec les versions précédentes, *ne devrait plus être utilisée*. Depuis la version 2.0, frenchb propose à la place une commande `\frenchbsetup{}`, à placer dans le préambule de chaque document (après le chargement de Babel), qui permet de faire un choix parmi toutes les options disponibles. La syntaxe est celle de l'extension `keyval`, largement utilisée par d'autres extensions comme `geometry` ou `hyperref`.

La commande `\frenchbsetup{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 : `\frenchbsetup{ShowOptions}` est équivalent à `\frenchbsetup{ShowOptions=true}`. L'option activée par défaut est indiquée entre crochets, 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 2), et que cette valeur est inversée sinon.

- `ShowOptions=true [false]` permet d'activer l'affichage dans le fichier `.log` des options disponibles, ce qui permet de les retrouver facilement.
- `StandardLayout=true [false*]` supprime toute action de frenchb sur la maquette : retour aux listes standard, pas de retrait des 1<sup>ers</sup> paragraphes des sections, notes de bas de page standard. Cette option remplace l'ancienne commande `\StandardLayout`.
- `GlobalLayoutFrench=false [true*]` peut servir, lorsque le français est la langue principale, à retrouver le comportement des versions de frenchb antérieures (pré-2.2) : dans les parties rédigées dans des langues autres que le français, la présentation des listes redevient standard et la mise en retrait des 1<sup>ers</sup> paragraphes des sections est supprimée. La présentation des notes de bas de page est toujours indépendante de la langue de travail (à la française ou standard pour tout le document selon la langue principale). Cette option remplace l'ancienne commande `\FrenchLayout`.
- `ReduceListSpacing=false [true*]` ; par défaut frenchb réduit les espaces verticaux<sup>9</sup> dans les toutes listes produites à partir de l'environnement `list`, en particulier les listes `enumerate`, `itemize` et `description`. On revient aux réglages standard (ceux de la classe utilisée) en choisissant l'option `ReduceListSpacing=false`. Cette option remplace l'ancienne commande `\FrenchListSpacingfalse`.
- `CompactItemize=false [true*]` ; par défaut frenchb redéfinit l'environnement `itemize` pour qu'aucun espace vertical ne soit ajouté à l'interligne standard entre les éléments d'une liste `itemize`.  
`CompactItemize=false` empêche cette redéfinition (ce qui peut être utile lorsqu'il y a un conflit avec `enumitem` par exemple). L'interligne entre les éléments des listes `itemize` est alors légèrement augmenté si `ReduceListSpacing=true` ou l'est nettement plus (on revient au réglage de bas de la classe utilisée) si `ReduceListSpacing=false` ; cette option remplace l'ancienne commande `\FrenchItemizeSpacingfalse`.
- `StandardItemLabels=true [false*]` restitue aux marqueurs des listes `itemize` les valeurs standard attribuées par la classe de document ou les extensions utilisés.

---

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

- `ItemLabels=\textemdash, \textbullet, \ding{43},...` [`\textendash*`]; cette option permet de choisir le marqueur utilisé dans les listes `itemize` en français (*sauf avec l'option* `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=\textemdash, \textbullet, \ding{43},...` [`\textendash*`]
  - `ItemLabelii=\textemdash, \textbullet, \ding{43},...` [`\textendash*`]
  - `ItemLabeliii=\textemdash, \textbullet, \ding{43},...` [`\textendash*`]
  - `ItemLabeliv=\textemdash, \textbullet, \ding{43},...` [`\textendash*`]
- `StandardLists=true` [`false*`] supprime toute action de `frenchb` sur les listes, elle équivaut aux trois options `StandardItemLabels=true`, `CompactItemize=false` et `ReduceListSpacing=false`. Lors de l'utilisation d'une classe ou d'une extension (`enumitem`, `paralist`, etc.) qui modifie la présentation des listes, il peut y avoir conflit avec `frenchb`; dans ce cas l'option `StandardLists` (ou éventuellement l'une ou l'autre des trois sous-options plus ciblées qu'elle regroupe) devrait régler le problème.
- `IndentFirst=false` [`true*`]; par défaut, `frenchb` applique un retrait (`\parindent`) identique pour tous les paragraphes, 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.
- `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.
- `AutoSpacePunctuation=false` [`true`]; par défaut, `frenchb` corrige la faute de saisie qui consiste à omettre l'espace devant la ponctuation haute (: ; ? !) en ajoutant automatiquement une espace insécable de taille adaptée. L'ajout d'espace n'a pas lieu lorsque la police courante est à espacement fixe : `\texttt{http://truc.muche}` produit `http://truc.muche`, mais `\textsf{http://truc.muche}` produit `http ://truc.muche` (espace parasite avant le « : »).  
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 de voir des espaces ajoutées à tort dans l'exemple ci-dessus en *sans-serif* ou dans « (!!) » par exemple.  
Une autre solution consiste à laisser le mode par défaut et à utiliser la commande `\string` pour éviter l'espace parasite le cas échéant : saisir (`\string!\string!\string!`). Le recours au codage `\string:` dans les URL peut être évité si on utilise la commande `\url` de l'extension `url` : `\url{http://}` produit toujours « `http://` » sans espace parasite.
- `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.
- `ThinSpaceInFrenchNumbers=true` [`false`] remplace le séparateur des milliers utilisé en français par la commande `\numprint{}` (ou son alias `\nombre{}`) pour le formatage des

nombre, par une espace fine (par défaut c'est une espace mot insécable en français). Cette option n'a d'effet que si l'extension `numprint` est chargée 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.

- `FrenchSuperscripts=false [true]` (option ajoutée en version 2.1) 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 une nouvelle commande `\fup` plus conforme aux usages francophones (voir ci-dessous, section 4.2).
- `LowercaseSuperscripts=false [true]` (option ajoutée en version 2.1) rend possible d'avoir des lettres supérieures en capitales (est-ce utile?). Par défaut, la nouvelle 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]` (option ajoutée en version 2.4) supprime les avertissements lorsque la commande `\makecaption` a été modifiée par une classe ou une autre extension et lorsque l'extension `bigfoot` est utilisée.
- `og=«`, `fg=»`<sup>10</sup>; 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 `\og` et `\fg`<sup>11</sup>. On peut ainsi saisir «guillemets» ou « guillemets » (avec ou sans espaces à l'intérieur) pour obtenir « guillemets » avec espaces insécables adéquates en français. En revanche *il ne faut pas coder explicitement les espaces insécables*: `«~guillemets~»` produirait « guillemets » où l'espace qui suit le guillemet ouvrant est trop large.

Les espaces insécables ne sont pas ajoutées lorsque la langue courante n'est pas le français afin de pouvoir saisir les guillemets allemands : »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.

- `OriginalTypewriter=true [false]`; par défaut, lorsqu'une police à espacement fixe est utilisée (mode verbatim, listings, etc.) `frenchb` n'ajoute aucun espace avant la ponctuation haute, ni après «' ni avant »' lorsque ces caractères sont activés dans `\frenchbsetup{}` (cf. ci-dessus). Cette option permet de supprimer ce comportement, par exemple pour recompiler à l'identique des anciens textes.
- `PartNameFull=false [true]`; par défaut `frenchb` numérote les parties créées par la commande `\part` en « Première partie », « Deuxième partie », etc. Ceci ne fonctionne pas correctement avec certaines classes, comme les classes AMS et SMF, qui redéfinissent la commande `\part`. L'option `PartNameFull=false` modifie la numérotation des parties en « Partie I », « Partie II », etc. ce qui supprime les risques de mauvais affichages du genre « Première partie I ».

**Ordre des options** – Il faut se souvenir que les options sont prises en compte dans l'ordre où elles sont écrites dans la commande `\frenchbsetup{}`. Exemple : un utilisateur souhaitant que `frenchb` 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 pourrait faire `\frenchbsetup{StandardLayout, IndentFirst}`. Si il codait

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

11. Toutefois, si des guillemets doivent être utilisés dans une commande `\index`, ils *doivent* être codés en `\og` et `\fg`.

`\frenchbsetup{IndentFirst,StandardLayout}` l'option `IndentFirst` serait annulée par `StandardLayout`. On obtiendrait également le résultat souhaité en codant `\frenchbsetup{StandardLists,FrenchFootnotes=false,AutoSpaceFootnotes=false}` là, l'ordre est indifférent car les options sont indépendantes.

## 4 Changements entre les versions 1.6 et 2.x

### 4.1 Principaux changements entre les versions 1.6 et 2.0

- Le support pour  $\text{\TeX}$ -2.09 et pour  $\text{\TeX}$ 2 $\epsilon$  en mode « compatible  $\text{\TeX}$ -2.09 » a été supprimé (la naissance de  $\text{\TeX}$ 2 $\epsilon$  date de 1994 !) mais le support pour les formats à base de Plain a été conservé. Les extensions `mltex` n'ont plus aucun intérêt en  $\text{\TeX}$  (avec les fontes 8-bits on utilise maintenant uniquement `pdfetex`), le support `mltex` a donc été supprimé en  $\text{\TeX}$ 2 $\epsilon$  mais il reste disponible en Plain $\text{\TeX}$ .
- La personnalisation du fonctionnement de `frenchb` devrait être faite uniquement à partir de la nouvelle commande `\frenchbsetup{}` dont les arguments suivent la syntaxe de `keyval`. Cette commande doit évidemment être placée uniquement dans le préambule et *après* le chargement de Babel. `\frenchbsetup{ShowOptions}` affiche dans le fichier `.log` la liste des options disponibles.  
Le lecteur est invité à se reporter à la section 3 p. 5 pour une description complète des options proposées, et à la section 4.8 p. 11 pour les problèmes de compatibilité entre les versions 1.6 et 2.x.  
Le fichier `frenchb.cfg`, conservé dans un souci de compatibilité avec les versions précédentes *ne devrait plus du tout être utilisé*.
- La commande `\frenchbsetup{StandardLayout}` débraie toutes les modifications faites par `frenchb` sur la maquette (listes, retrait des 1<sup>ers</sup> paragraphes de sections et notes de bas de page). Toutes les redéfinitions concernant la maquette, faites dans le préambule avant ou après le chargement de `babel/frenchb`, sont prises en compte ; ceci devrait faciliter la vie des auteurs de classes LaTeX.

### 4.2 Changements entre les versions 2.0 et 2.1

Le seul changement affecte la commande `\up` et ses dérivées (`\ieme`, `\no`, `\No`, `\primo`, etc.). Auparavant `\up` était un raccourci de la commande LaTeX `\textsuperscript` conçue en fait pour produire des exposants en mode texte (appel de note par exemple) plutôt que des lettres supérieures.

`\up` est désormais un alias de `\fup`, nouvelle commande qui utilise les vraies lettres supérieures si elles existent dans la police courante et les simule sinon. Ceci fonctionne pour l'instant

- sous XeLaTeX à condition de charger les extensions `fontspec` et `realscripts` et d'utiliser une police de type OpenType qui connaisse la directive « `VerticalPosition=Superior` » (polices Linux Libertine O, Iwona, par exemple) ;
- sous (pdf)LaTeX avec certaines polices type 1 expertes (Fourier-Utopia).

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

Exemple : en codant `M\up{me} M\up{lle} M\up{gr}` on obtient maintenant M<sup>me</sup>, M<sup>lle</sup>, M<sup>gr</sup>, à comparer au résultat de `\textsuperscript Mme, Mlle, Mgr`. Avec les vraies supérieures de la même police (Utopia-expert, type 1) on obtiendrait M<sup>me</sup>, M<sup>lle</sup>, M<sup>r</sup> (le « g » est avalé car absent des supérieures d’Utopia-expert!). Pour pallier ce type de problème (la police OpenType Jenson Pro n’a pas non plus de « g supérieur », elle affiche `M\up{gr}` sous la forme M<sup>gr</sup>), une variante étoilée de `\up` est fournie. `M\up*{gr}` force l’utilisation de supérieures simulées même si de vraies supérieures sont disponibles. La commande étoilée `\up*` n’est à utiliser qu’en cas de problème avec `\up`.

La différence entre `\fup` et `\textsuperscript` est plus visible en grande taille :

M<sup>me</sup>, M<sup>lle</sup>, M<sup>gr</sup> (`\fup`) et M<sup>me</sup>, M<sup>lle</sup>, M<sup>gr</sup> (`\textsuperscript`).

Il est possible de retrouver l’ancien comportement de `\up` en ajoutant dans le préambule `\frenchbsetup{FrenchSuperscripts=false}`. Cette option est sans effet sur `\up*`.

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`).

### 4.3 Changements entre les versions 2.1 et 2.2

À partir de la version 2.2, `frenchb` ne modifie la mise en page globale du document *même en français* que si l’option `french` est la *dernière* langue chargée par Babel, ce qui en fait la langue principale du document.

Si `french` est la *dernière* langue chargée par Babel, *tout le document*, y compris les passages rédigés dans une langue autre que le français, adopte la mise en page française (listes, notes de bas de page et retrait des premiers paragraphes des sections).

En revanche, si `french` n’est pas la dernière langue chargée par Babel, la mise en page standard (celle définie par la classe et les extensions chargées) est respectée en ce qui concerne les listes, les notes de bas de page et le retrait des premiers paragraphes des sections.

### 4.4 Changements entre les versions 2.2 et 2.3

À partir de la version 2.3, `frenchb` n’insère plus d’espace devant la ponctuation haute lorsque la fonte courante est à espacement fixe (`\texttt{}`, `\ttfamily`), ceci même si l’option `AutoSpacePunctuation` est active (`true`). Cette modification devrait faciliter la saisie des chemins MS-DOS ou des URL : `\texttt{C:/}` produit désormais C:/ sans avoir recours à `\string:`.

Il est possible de revenir à l’ancien comportement de `frenchb` en ajoutant l’option `OriginalTypewriter` dans `\frenchbsetup{}`.

Autre changement : il n’y a plus d’erreur à la compilation lorsqu’on code `2\up{ième}`, *mais* – soit on utilise une police sans vraies lettres supérieures et on obtient « 2<sup>ième</sup> » ce qui *n’est pas correct* (l’abréviation de deuxième est 2<sup>e</sup>), – soit on utilise une police disposant de vraies lettres supérieures et le « è » disparaît sans prévenir : on obtient « 2<sup>ime</sup> » car il n’y *jamais* de diacritiques dans les lettres supérieures.

Vous voilà prévenus !

De nouvelles commandes facilitent la personnalisation des notes de bas de page. 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, tout comme le retrait des notes par rapport à la marge gauche (voir page 3).

## 4.5 Changements entre les versions 2.3 et 2.4

Les motifs de césures étant maintenant codés en Unicode, il a été nécessaire d'adapter les commandes `\extrasfrenchb` et `\noextrasfrenchb` pour prendre en compte le nouveau codage de l'apostrophe.

L'option `SuppressWarning` (voir section 3) a été ajoutée.

La compatibilité avec l'extension `enumitem` a été améliorée : l'option `CompactItemize=false` est activée automatiquement si `enumitem` est chargé (v. 2.4b).

Lorsqu'on utilise les options `og=«` et `fg=»` de `\frenchbsetup{}`, les espaces ajoutés automatiquement après «' et avant '» ne le sont plus lorsque la fonte courante est à espacement fixe (`\texttt{}`, `\ttfamily`), donc en particulier en mode verbatim. Il est possible de revenir à l'ancien comportement de `frenchb` en ajoutant l'option `OriginalTypewriter` dans `\frenchbsetup{}`. Ainsi le traitement des caractères «' et '» est homogène avec celui de la ponctuation haute.

## 4.6 Changements entre les versions 2.4 et 2.5

Le principal changement est l'abandon du recours aux caractères actifs pour la gestion de la ponctuation haute lorsqu'un format basé sur XeTeX est utilisé (XeLaTeX par exemple). Le mécanisme des caractères actifs est remplacé par celui des `\XeTeXinterchartoks` qui permet d'insérer une commande entre deux caractères d'un texte source sans que ceux-ci soient rendus actifs.

Les caractères `:` `;` `!` `?` restent actifs en (La)TeX standard.

Il n'y a aucun changement visible dans l'interface utilisateur : la taille des espaces est toujours paramétrable (option `ThinColonSpace` ou redéfinition de `\Fcolonspace` ou `\Fthinspace`) et l'option `AutoSpacePunctuation=false` a le même effet en XeLaTeX qu'en LaTeX.

Une nouvelle commande `\NoAutoSpacing` a été ajoutée pour généraliser l'effet de `\shorthandoff{; ! ?}` ; elle fonctionne aussi bien en LaTeX qu'en XeLaTeX. Elle peut être utilisée à chaque fois que survient un problème dû aux caractères actifs ou que des espaces indésirables apparaissent en sortie (avec les guillemets dans les listings par exemple). `\NoAutoSpacing` s'utilise *dans un groupe* comme ceci :

```
{\NoAutoSpacing <texte à protéger>}
```

Un bug gênant a été corrigé : `frenchb` ne perturbe plus les commandes `\frenchspacing` et `\nonfrenchspacing` dans les autres langues que le français.

## 4.7 Incompatibilités assumées

J’ai profité du passage à la version 2.0 pour revenir sur certains choix qui me paraissent mauvais à l’usage :

- Les légendes des figures et des tableaux a changé : les abréviations « Fig. » et « Tab. » ont été changées en « Figure » et « Table ». Motivation : les abréviations Fig. et Tab. sont tolérables dans les références à une figure ou à un tableau, mais pas dans les légendes elles-mêmes.
- En LaTeX, la commande `\nombre`<sup>12</sup> fait désormais appel à l’extension `numprint.sty`, qu’il faut charger *après* Babel et avec l’option `autolanguage` si on veut que le formatage des nombres dépende de la langue courante :

```
\usepackage[autolanguage]{numprint}
```

Motivation : l’extension `numprint.sty` fait un meilleur travail et offre plus de possibilités (voir la documentation de cette extension).

- La commande `\bsc` ne fait plus appel à une `\hbox` pour empêcher la coupure des noms qu’elle contient mais à un `\kern0pt`. Motivation : le recours à `\hbox` gêne le fonctionnement `microtype` (la largeur de la boîte est figée). Ce changement fait aussi que les noms composés peuvent maintenant être coupés (seulement) à l’emplacement du trait d’union.
- Lorsque le français est la langue principale du document (dernière langue chargée par babel), les notes de bas de page sont présentées par défaut « à la française » (c.-à-d. selon les préconisations de l’Imprimerie nationale) : une espace fine précède l’appel de note et les notes elles-mêmes sont numérotées « 1. », « 2. », etc. et plus « <sup>1</sup> », « <sup>2</sup> ». Le résultat est le même que celui qui était obtenu avec la version 1.6 *et* les commandes `\AddThinSpaceBeforeFootnotes` et `\FrenchFootnotes`. Motivation : ces commandes étaient largement ignorées par les utilisateurs.

- Depuis la version 2.2, les préambules suivants ne donnent plus le même résultat !

```
\usepackage[english,french]{babel}      \usepackage[french,english]{babel}
\begin{document}                        \begin{document}
                                         \selectlanguage{french}
```

Avec celui de gauche la langue principale est le français, les notes de bas de page, les listes seront présentées à la française dans tout le document, et les premiers paragraphes des sections seront tous présentés avec en retrait, comme les suivants ceci quelle que soit la langue courante.

Avec celui de droite, la mise en page sera celle définie par la classe, `frenchb` ne la modifie plus, la langue principale du document étant l’américain.

Motivation : avoir, dans un même document, des listes ou des notes de bas de page présentées différemment selon la langue est perturbant pour la lecture. Une unification de la présentation en fonction de la langue principale me semble préférable.

## 4.8 Comment recompiler à l’identique un document LaTeX écrit pour `frenchb 1.6` ?

Il suffit d’ajouter dans le préambule du document, après le chargement de Babel :

```
\renewcommand*{\bsc}[1]{\leavevmode\hbox{\scshape #1}}
\addto\captionsfrench{\def\figurename{\scshape Fig.}}
\addto\captionsfrench{\def\tablename{\scshape Tab.}}
```

---

12. En Plain, elle ne formate plus les nombres et émet un message d’avertissement.

```
\frenchbsetup{GlobalLayoutFrench=false,StandardLayout=false,
              AutoSpaceFootnotes=false,FrenchFootnotes=false,
              FrenchSuperscripts=false,OriginalTypewriter=false}
\usepackage[autolanguage]{numprint}
```

Ces commandes devront être adaptées aux besoins : le chargement de `numprint` est inutile si la commande `\nombre` n'est pas utilisée, les arguments de la commande `\frenchbsetup` devront être adaptés si nécessaire (voir section 3). Si le document est intégralement en français, les options `GlobalLayoutFrench=false` et `StandardLayout=false` sont inutiles, si `french` est la dernière option de Babel, la première option suffit, sinon la seconde suffit.

Ceux qui ont besoin de créer des fichiers compilables avec les deux versions (1.6 et 2.x), peuvent le faire en ajoutant à leur préambule un test vérifiant l'existence ou non de la commande `\frenchbsetup` :

```
\makeatletter
\@ifundefined{frenchbsetup}{% frenchbsetup SANS \ ici !
  % \frenchbsetup{...}
}{%
  % anciennes commandes
}
\makeatother
```

## 5 Problèmes avec les quatre caractères actifs (; : ! ?)

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 (`xypic`, `xcolor`, `arabtex` 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 `frenchb` !

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`<sup>13</sup> *dans un groupe* comme ceci :

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

Il est également possible, *sauf sous XeLaTeX*, de désactiver un seul ou plusieurs caractères actifs grâce à la commande `\shorthandoff` de Babel. Le mieux est de le faire *localement* comme ceci :

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

---

13. Elle fonctionne à la fois en (La)TeX et en Xe(La)TeX.

## 6 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 « collection-langfrench » (TeXLive, MacTeX) ou similaire selon votre distribution. 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.

## 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 frenchb (; ! ?) peuvent perturber certaines extensions, c'est le cas de [xypic](#), [xcolor](#), [arabtex](#) par exemple, voir comment y remédier section 5.  
En règle générale, il vaudrait mieux éviter les « : » dans les `\label`, les remplacer systématiquement par des tirets « - » est une saine précaution. Si on tient aux « : » dans les `\label` et si on utilise [natbib](#), il convient de charger [natbib](#) *avant* `babel`.  
La présence du caractère « : » dans les `\label` peut provoquer des erreurs lorsque l'extension [listings](#) est utilisée. Il semble qu'il suffise de charger [listings](#) *après* `babel` pour que le problème disparaisse.
- 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)<sup>14</sup> 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 encadrer les commandes du type `\title` entre un `\shorthandon{; ! ?}` (avant) et un `\shorthandoff{; ! ?}` (après), si on tient à les laisser dans le préambule.
- frenchb modifie la présentation des listes ce qui peut perturber les classes ou extensions qui veulent également le faire. Dans ce cas, ou si on veut soi-même agir sur la présentation des listes, il convient de débrayer l'action de frenchb en utilisant la commande `\frenchbsetup{}` avec les options adéquates (cf. section 3) dans le préambule du document (après le chargement de Babel).
- Certaines classes ([amsbook.cls](#), [smfbook.cls](#), [beamer.cls](#),...) redéfinissent `\part`, ce qui peut provoquer des titres du genre « Première partie I ». Parade : ajouter l'option `PartNameFull=false` dans `\frenchbsetup{}`.

---

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

- L’option `multiple` de l’extension `footmisc` insère normalement une virgule entre les appels de notes multiples. Pour que ce mécanisme fonctionne avec `frenchb` il faut ajouter l’option `\frenchbsetup{AutoSpaceFootnotes=false}`, sinon `frenchb` remplace la virgule par une espace fine.
- Dans une commande `\index{}`, les guillemets français doivent obligatoirement être codés sous la forme `\og, \fg` : `\index{\og toto\fg}` 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 avec BibTeX

Ce qui suit ne concerne que les bibliographies créées avec BibTeX (*pas* celles faites « à la main » dans l’environnement `thebibliography`).

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>15</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 adapter le fichier `babelbst.tex` aux langues utilisées.

## 9 Compatibilité avec FrenchPro

Il est souhaitable qu’un texte saisi avec FrenchPro de Bernard GAULLE puisse être compilé avec un minimum de modifications sur une machine utilisant Babel et `frenchb` et réciproquement.

En ce qui concerne les guillemets français, FrenchPro rend actifs les caractères `<` et `>` afin de saisir les guillemets sous la forme `<<` et `>>` tandis que `frenchb` s’y refuse et utilise en interne des macros `\og` et `\fg`.

Lorsqu’on travaille en codage T1 avec `frenchb`, 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

15. 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).

dans `\frenchbsetup{}`) sont préférables : elles introduisent automatiquement des espaces insécables et plus fines pour un meilleur rendu typographique.

Les points de suspensions sont saisis `...` avec FrenchPro et `\dots` avec frenchb.

Les commandes suivantes peuvent être ajoutées au préambule pour émuler certaines commandes de FrenchPro :

```
\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 FrenchPro facilitant sa compilation avec frenchb. L'appel à FrenchPro est remplacé en un appel à Babel/frenchb, les guillemets `<<` et `>>` sont convertis en `\og` et `\fg` et les `...` en `\dots`, enfin quelques commandes spécifiques à FrenchPro 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>