Le format de dictionnaire numérique LING

et son format passerelle PRELING

Spécification 1.1

Edition **10**

10/01/2011

1. Généralités

Le format LING est un format de fichier utilisable pour les dictionnaires numériques et les documents apparentés.

La présente spécification et ses variantes et évolutions directes sont publiques et librement utilisables. Aucun droit de propriété intellectuelle ou industrielle ne peuvent y être attachés.

Pourquoi ce format ?

Les formats de dictionnaires numériques sont nombreux...

L'une des raisons d'être du format LING est essentiellement de remédier à certaines insuffisances des formats existants en matière de gestion des relations internes entre les entrées d'un même dictionnaire ou des relations externes entre plusieurs dictionnaires. C'est dans cette optique que le format LING intègre une table d'identifiants "wordID" pointant sur les éléments du dictionnaire et *indépendants de leur conten*u. L'établissement de relations dans le dictionnaire ou entre dictionnaires est ainsi facilité, tout en permettant l'édition des libellés des entrées du dictionnaire sans rupture des liens relationnels. Une autre particularité du format LING est d'être un format binaire conçu parallèlement à un format texte lui servant de passerelle d'importation/exportation ce qui en facilite donc l'édition.

Conventions de ce document :

```
Contenu du fichier  # commentaire non inclus dans le fichier
```

<00> symbolise le zéro binaire ("caractère" d'ordinal 0, binaire 00000000, hexadécimal 00 00), à ne pas confondre avec le caractère "0" (ordinal 48) !

2. Le fichier LING

L'extension conseillé pour les noms des fichier LING est *.ling

Afin d'assurer une bonne compatibilité multiplateforme de ces fichiers de dictionnaires, les caractères accentués, les majuscules et les espaces sont fortement déconseillés dans les noms de fichiers :

Exemples :

francais_suedois.ling : OUI

Français – Suédois.ling : NON

3. Structure interne d'un fichier LING

Un fichier Ling est un fichier **binaire**. Il ne peut donc être édité à l'aide d'un éditeur de texte. Pour pallier à ceci, et permettre l'édition globale du dictionnaire, un format texte passerelle avec le Format LING, du nom de PRELING, est également défini plus bas.

Toutes les données textuelles incluses dans un fichier LING sont encodées en **Utf-8**.

Un fichier Ling est divisé en blocs. Le fichier débute par un bloc de cartographie globale du fichier, *l'ordre des autres blocs dans le fichier est donc sans importance*, seul ce bloc de cartographie a une place précise au début du fichier.

Un fichier LING est composé des blocs suivants :

- Six blocs standards obligatoires

- Bloc identifiant et de cartographie du fichier (en-tête du fichier)
- Bloc des propriétés du dictionnaire
- Bloc des entrées du dictionnaire
- Bloc des wordIDs des entrées du dictionnaire
- Bloc de cartographie des notices
- Bloc des notices

- Deux blocs standards facultatifs :

- Bloc de l'image 1 (icône de la langue source)
- Bloc de l'image 2 (icône de la langue cible)

- Des blocs non standards facultatifs :

- …
- …

Des blocs non standards propres à l'utilisateur ou à certains logiciels peuvent être librement ajoutés en nombre illimité. L'accès aux blocs du fichier LING se faisant en utilisant les coordonnées fournies par le bloc de cartographie du fichier, les blocs additionnels non standards seront ainsi ignorés par les logiciels ne les prenant pas en compte.

4. Le Bloc identifiant et cartographie du fichier

Les 14 premiers octets/caractères du fichier servent d'identifiant pour reconnaître un fichier LING valide et indiquent la version de la spécification LING suivie :

Le reste du bloc contient la cartographie du fichier : une suite de nombres encodés sur 32 bits (4 octets), octet lourd à gauche.

La taille standard de ce bloc identifiant et cartographie est donc toujours de 70 octets :

Identifiant de type de fichier	=	14 octets
Cartographie : 14 x 4 octets	=	56 octets
Total	=	70 octets

Ce bloc peut cependant être allongé librement, au-delà des 70 premiers octets, si cela est nécessaire à un usage propre à certains logiciels, à condition que la partie additionnelle suive directement la partie standard sans modification structurelle de celle-ci.

Les coordonnées de cartographie du fichier sont ordonnées de la manière suivante :

```
%ling/01.01.00   # octets 1-14 : Identifiant de fichier et version LING
XX XX            # octets 15-18 : Offset du bloc des Propriétés
XX XX            # octets 19-22 : Taille du bloc des Propriétés
XX XX            # octets 23-26 : Offset du bloc des Entrées
XX XX            # octets 27-30 : Taille du bloc des Entrées
XX XX            # octets 31-34 : Offset du bloc des wordIDs
XX XX            # octets 35-38 : Taille du bloc des wordIDs
XX XX            # octets 39-42 : Offset du bloc Cartographie des Notices
XX XX            # octets 43-46 : Taille du bloc Cartographie des Notices
XX XX            # octets 47-50 : Offset du bloc des Notices
XX XX            # octets 51-54 : Taille du bloc des Notices
XX XX            # octets 55-58 : Offset du bloc Image1 (00 00 si absent)
XX XX            # octets 59-62 : Taille du bloc Image1 (00 00 si absent)
XX XX            # octets 63-66 : Offset du bloc Image2 (00 00 si absent)
XX XX            # octets 67-70 : Taille du bloc Image2 (00 00 si absent)
```

Nb : les octets ci-dessus sont numérotés à partir de 1

5. Le Bloc des propriétés du dictionnaire

Ce bloc est constitué d'une suite de champs textuels encodés en utf-8 et séparés par <00>.

L'ordre des champs de propriété dans le bloc est sans importance.

Tous les champs de propriété sont *facultatifs*, y compris les champs standards définis plus bas. Un fichier LING dont le bloc des propriétés est vide est valide.

Un bloc de propriétés est constitué de propriétés standards et éventuellement de propriétés additionnelles.

Propriétés standards :

Leur présence est *facultative* mais leur liste est *limitative*. Leur libellé et leur usage sont strictement codifiés. Voir plus bas.

Les logiciels devront si besoin affecter une valeur par défaut aux propriétés non explicitement incluses dans le fichier LING mais ne devront pas générer d'erreur en ce cas.

Propriétés additionnelles :

Leur présence est *facultative* et leur nombre est *illimité*.

Les propriétés additionnelles sont des champs non standards, c'est-à-dire des champs de propriétés absents de la présente spécification.

Ces champs propres à l'auteur, à l'utilisateur ou à certains logiciels peuvent être librement définis et ajoutés dans le bloc des propriétés.

Le nom d'une propriété additionnelle est libre mais il DOIT être préfixé par "x_ling_" (ceci afin d'éviter des collisions de noms entre propriétés standards et propriétés additionnelles) :

```
x_ling_MonChampPerso="ma valeur"
```

En conséquence, un fichier LING contenant une propriété absente de la liste des propriétés standards et dont le nom n'est pas préfixé par "x_ling_" est invalide .

Les logiciels ne reconnaissant pas une propriété additionnelle doivent l'ignorer tout en la respectant, c'est-à-dire la conserver lors de toute édition (modification) du fichier du dictionnaire.

5.1 Syntaxe

Le nom du champ (le nom de la propriété) est séparé de sa valeur par "=".

La liste des noms des champs standards est indiquée plus bas. Les noms des champs additionnels doivent être préfixés par "x_ling_".

La casse (majuscule/minuscule) du nom des champs doit être respectée.

```
nomChampStandard=<valeur compatible>
x_ling_MonChampPerso="ma valeur"
```

Les valeurs peuvent être de quatre types :

- Valeur textuelle littérale simple
- Liste de valeurs textuelles
- Valeur Booléenne
- Valeur numérique

5.1.1 Valeur textuelle simple :

Indiquée par (Txt) dans la liste des champs ci-dessous

Les valeurs textuelles (chaînes) sont entourées de guillemets (doubles si possible, simples si nécessaire). Les éventuels guillemets doubles internes au texte doivent être inscrits sous le forme de l'entité Html `&quot;` :

Dans les contenus des champs de propriétés destinés à l'affichage, les balises suivantes sont autorisées:

** **	: saut de ligne.
...	: texte en gras.
...	: texte en italique.
<u>...</u>	: texte souligné.
...	: diminuer la taille du texte.
<big>...</big>	: augmenter la taille du texte.

```
nomChamp="Texte du <i>champ</i> ligne 1<br>Texte du <b>champ</b> ligne 2"
```

La longueur des données textuelles de chaque champ n'est, sauf cas particulier mentionné, pas limitée.

5.1.2 Liste de valeurs textuelles :

Indiquée par (Txt-lst) *dans la liste des champs ci-dessous*

Les champs de type liste de chaînes sont segmentés par des virgules et chaque chaîne est entre guillemets (doubles si possible, simples si nécessaire). Les éventuels guillemets doubles internes au texte doivent être inscrits sous le forme de l'entité Html `&quot;` :

```
nomChamp="...","...","..."
```

5.1.3 Valeur booléenne :

Indiquée par (Bool) *dans la liste des champs ci-dessous*

Les valeurs booléennes sont représentées par les mentions `True` (Vrai) et `False` (faux), inscrits *sans* guillemets et avec une majuscule :

```
nomChamp=True
```

5.1.4 Valeur numérique :

Indiquée par (Num) *dans la liste des champs ci-dessous*

Les valeurs numériques sont inscrites en notation décimale, *sans* guillemets :

```
nomChamp=123
```

5.2 La liste des champs des propriétés

Voir plus bas pour la présentation détaillée de chaque champ de propriété.

Champs standards :

```
minCompatVersion="..."       # (Txt) Compatibilité LING minimale
maxCompatVersion="..."       # (Txt) Compatibilité LING maximale
dicName="..."                # (Txt) Nom convivial du dictionnaire
langName1="..."              # (Txt) Nom convivial de la langue 1
langName2="..."              # (Txt) Nom convivial de la langue 2
langIso1="..."               # (Txt) Code ISO 639-2 de la langue 1
langIso2="..."               # (Txt) Code ISO 639-2 de la langue 2
langNameUser="..."           # (Txt) Nom de la langue de l'utilisateur
langIsoUser="..."            # (Txt) Code ISO 639-2 langue utilisateur
langFamily1="..."            # (Txt) Famille linguistique de la langue 1
langFamily2="..."            # (Txt) Famille linguistique de la langue 2
isReverseDic= True|False     # (Bool) est un dictionnaire inverse auto
doReverseDic= True|False     # (Bool) est un dico conçu pour être inversé
reverseDicFileName="..."     # (Txt) Nom fichier du dico inverse/orig.
reverseDicName="..."         # (Txt) Nom convivial du dico inverse/orig.
sortEquPatterns="...","..."  # (Txt-lst) Motifs d'équivalence de tri
sortEquPatternsRev="..","..."# (Txt-lst) Equ. de tri (dico inverse)
wordcount=...                # (Num) Nombre d'entrées du dictionnaire
mainAuthors="...","..."      # (Txt-lst) Auteur(s) principaux du dico
altAuthors="...","..."       # (Txt-lst) Auteur(s) accessoires
contactAuthor="..."          # (Txt) Contact des auteurs
shortAuthors="..."           # (Txt) auteurs, formulation courte
```

```
dicStatus="..."              # (Txt)Statut et licence du dictionnaire
showDicStatus= True|False    # (Bool) Afficher le statut du dictionnaire
copyright="..."              # (Txt) Mention de copyright ou équivalent
creationDate="..."           # (Txt) Date de création du dictionnaire
versionDate="..."            # (Txt) Date de la présente version du dico
localEditDate="..."          # (Txt) Date d'édition locale
dicID="..."                  # (Txt) Identifiant du dictionnaire
dicVersionNumber="..."       # (Txt) N° de version du dictionnaire
dicUrl="..."                 # (Txt) URL du dictionnaire sur le Web
verUrl="..."                 # (Txt) URL d'un éventuel fichier annexe
dicInfo="..."                # (Txt) Texte de présentation du dico
showDicInfo= True|False      # (Bool) Afficher les infos
protected1="..."             # (Txt) Protocole de protection des entrées
protected2="..."             # (Txt) Protocole de protection des notices
displayFontName1="..."       # (Txt) Police d'affichage de la langue 1
displayFontName2="..."       # (Txt) Police d'affichage de la langue 2
grammarEncoding1="..."       # (Txt) Norme codage grammatical langue 1
compatPlugins="...","..."    # (Txt-lst) Greffons compatibles
noCompatPlugins="...","..." # (Txt-lst) Greffons incompatibles
usePlugins="...","..."       # (Txt-lst) Greffons à associer au dico
wordGroups="...","..."       # (Txt-lst) IDs des groupes de mots du dico
biblio="...","..."           # (Txt-lst) Références biblio du dico
showBiblio=True|False        # (Bool) Afficher les références biblio
extFieldCount=...            # (Num) Nbre champs d'extension des notices
extFieldList="...","..."     # (Txt-lst) Noms des champs d'extension
```

Champs de propriété additionnels (propres à l'utilisateur ou au logiciel) **:**

```
x_ling_<prop1>="..."         # (Txt, Bool, Num. ou Txt-lst)
x_ling_<prop2>="..."         # (Txt, Bool, Num. ou Txt-lst)
...
```

5.3 [minCompatVersion] et [maxCompatVersion]

Format : Texte (entre guillemets).

Ces champs contiennent les numéros de la spécification LING minimale et maximale en deçà ou au delà de laquelle le dictionnaire n'est plus compatible.

Ces champs définissent donc un *intervalle* de compatibilité du dictionnaire. La version LING *nominale* est indiquée dans le bloc identifiant et cartographie (cf.). Ces trois valeurs peuvent être identiques.

Si l'un ou l'autre de ces champs est absent ou non complété ou les deux, la valeur du champ manquant est considérée comme étant identique à la version LING nominale du fichier.

Syntaxe : trois blocs de 2 chiffres séparés par des points "00.00.00" comme pour la version LING nominale du bloc identifiant, suivis éventuellement d'un espace et d'un texte quelconque "00.00.00 xxxxx".

Exemple :

```
minCompatVersion="01.00.00"
maxCompatVersion="02.04.00 beta"
```

5.4 [dicName]

Format : Texte (entre guillemets).

Ce champ contient le nom convivial du dictionnaire, c'est-à-dire le nom du dictionnaire à destination de l'usager, tel qu'il s'affichera dans l'interface des logiciels.

Ce nom peut contenir des caractères spéciaux et accentués (contrairement au nom de fichier du dictionnaire). Dans le cas de dictionnaires bilingues, il est classiquement constitué des noms des langues séparés par un tiret.

Exemples :

```
dicName="Français - Suédois"
```

```
dicName="Dictionnaire de termes de marine"
```

5.5 [langName1] , [langName2]
[langIso1] , [langIso2]

Format : Texte (entre guillemets).

Les champs *langName* contiennent les noms conviviaux des langues du dictionnaire.

La *Langue 1* est la langue source, la *Langue 2* est la langue cible. "Langue" est ici à prendre au sens large et peut être interprété comme "thème" ou "sujet" dans le cas d'un dictionnaire monolingue.

Les champs *langIso* contiennent les codes des langues du dictionnaire suivant la norme ISO 639. Voir http://fr.wikipedia.org/wiki/ISO_639

Cette norme ISO ayant plusieurs variantes, la norme suivie doit être indiquée par un préfixe suivie de ":". En l'absence de ce préfixe, la norme utilisée sera considérée être la norme ISO 639-2. Voir http://fr.wikipedia.org/wiki/Liste_des_codes_ISO_639-2

La norme ISO prévoyant des codes pour les cas de langues indéterminées ou multiples. Il n'y a donc que peu de raisons de laisser ces champs vides.

Les deux codes ISO devraient donc être identiques dans le cas d'un dictionnaire monolingue.

Exemples :

```
langName1="Français"
langName2="Suédois"
langIso1="639-2:fra"
langIso2="639-2:swe"
```

```
langName1="Termes de marine"
langName2="Définitions"
langIso1="639-2:fra"
langIso2="639-2:fra"
```

5.6 [langNameUser] , [langIsoUser]

Format : Texte (entre guillemets).

Ces champs contiennent le nom convivial (*langNameUser*) et le code ISO639 (*langIsoUser*) de la langue de l'utilisateur supposé du dictionnaire, qui peut parfois être une langue différente des langues utilisées dans le dictionnaire. Il s'agit donc de *la langue de celui à qui est destiné le dictionnaire*.

Voir les indications qui précèdent, à propos des champs *langNameX* et *langIsoX* pour l'usage et la syntaxe, qui sont identiques.

Les logiciels peuvent utiliser ce champ, s'il est complété, pour, par exemple, adapter leur interface ou modifier leur présentation typographique en fonction des usages de la langue de l'utilisateur.

5.7 [langFamily1] , [langFamily2]

Format : Texte (entre guillemets).

Ces champs contiennent les noms des familles linguistiques des langues 1 et 2.

Syntaxe : libellé libre ou, de préférence, en référence à une norme (ISO ou autre), en indiquant le nom de la norme par un préfixe suivi de ":".

Une hiérarchie de plusieurs familles peut être indiquée par "*". Une conjonction non hiérarchique peut être indiquée par "+".

Exemples :

```
langFamily1="latine"
```

```
langFamily2="scandinave"
```

```
langFamily1="ISO 639-5:ine*itc*roa"
langFamily2="ISO 639-5:ine*gem"
```

5.8 [isReverseDic]

Format : Booléen.

Si sa valeur est *True* (Vrai) ce champ indique que le dictionnaire est le résultat d'une inversion automatisée et non un dictionnaire rédigé humainement.

Le logiciels réalisant une inversion automatique d'un dictionnaire source doivent obligatoirement mettre la valeur de ce champ à *True* dans le dictionnaire inversé.

5.9 [doReverseDic]

Format : Booléen.

Si sa valeur est *True* (Vrai) ce champ indique que le dictionnaire a été rédigé en prenant en compte le fait qu'il puisse ensuite être inversé par un processus automatisé.

L'absence de ce champ dans le dictionnaire équivaut à doReverseDic=False.

Ce champ peut être utilisé par les logiciels soit pour interdire l'inversion d'un dico marqué comme ne pouvant pas être inversé, soit, au minimum, pour avertir l'utilisateur du risque d'un résultat à la fiabilité médiocre et insatisfaisante s'il inverse ce dictionnaire.

Après un processus d'inversion automatique, la valeur de ce champ doit obligatoirement être mis à *False* par les logiciels dans le dictionnaire inversé.

Remarque : ce champ est un indicateur au niveau global du dictionnaire. Il est également possible d'affiner les indications d'inversion au niveau de chaque entrée du dictionnaire. Voir l'attribut "r" du bloc des données > champs des attributs.

5.10 [reverseDicFileName]

Format : Texte (entre guillemets).

Ce champ contient le nom du fichier du dictionnaire inverse.

Nb : il n'est pas obligatoire que le dictionnaire inverse soit au même format que le dictionnaire courant.

Les logiciels doivent considérer que *le fait que ce champ soit complété ne signifie pas que ce fichier existe physiquement*. Le contenu de ce champ est alors à traiter comme une proposition de nom de fichier, qui sera remplacée par le nom réel après l'inversion, si l'utilisateur a choisi un nom différent lors du processus d'inversion. L'auteur d'un dictionnaire a toujours intérêt à proposer dans ce champ un nom de fichier pour le dictionnaire inverse, quand son dictionnaire source a été conçu pour être inversable.

Ce nom de fichier peut être un chemin relatif au fichier du dictionnaire mais *jamais un chemin absolu* ! Sans chemin relatif indiqué, le fichier du dictionnaire inverse devra être recherché par les logiciels dans le même répertoire que le fichier du dictionnaire courant. En cas de chemin relatif, le séparateur des éléments du chemin est le "/", quel que soit la plateforme.

Exemple :

```
doReverseDic=True
reverseDicFileName="suedois_francais.ling"
```

5.11 [reverseDicName]

Format : Texte (entre guillemets).

Ce champ contient une proposition de nom convivial pour le dictionnaire inverse quand celui-ci doit être secondairement construit par un processus automatisé.

Un auteur a toujours intérêt à proposer ainsi un nom convivial pour le dictionnaire inverse quand le dictionnaire source a été conçu pour être inversable.

Exemple :

```
dicName="Français - Suédois"
reverseDicName="Suédois - Français"
```

5.12 [sortEquPatterns] [sortEquPatternsRev]

Format : Liste de textes entre guillemets séparés par des virgules "...","...","...","..."

Le champ *sortEquPatterns* contient la liste des équivalences de tri qui devront être utilisées pour trier alphabétiquement le dictionnaire. Chaque dictionnaire peut ainsi spécifier en interne la façon dont il doit être trié par les logiciels.

Le champ *sortEquPatternsRev* contient la liste des équivalences de tri qui devront être utilisées pour trier alphabétiquement un éventuel dictionnaire inverse automatique. Formulé autrement, le contenu du champ *sortEquPatternsRev* correspond au contenu du champ *sortEquPatterns* du dictionnaire inverse et réciproquement.

Syntaxe : <motif de caractères du dico>:<motif de caractères équivalents>

Quelques symboles sont utilisables :

"%%" correspond à l'équivalence "absence de caractère(s)".

"$$" correspond à l'espace.

L'usage de ces motifs évite des erreurs de saisie et protège les chaînes contre des formatages intempestifs de la part des logiciels.

Exemple :

```
sortEquPatterns="ö:oe","æ:ae","-:$$","...:%%"
```

Dans cet exemple...

- "ö" sera trié comme s'il s'agissait de "oe"
- "æ" sera trié comme s'il s'agissait de "ae"
- l'écriture d'un mot avec ou sans trait d'union sera équivalente
- les points de suspension seront ignorés

5.13 [wordcount]

Format : valeur numérique.

Ce champ contient le nombre d'entrées présentes dans le dictionnaire.

Compléter ce champ permet aux logiciels de dépister des données corrompues ou un problème au chargement si le nombre d'entrées effectivement chargées diffère du nombre inscrit.

5.14 [mainAuthors]

Format : Liste de textes entre guillemets séparés par des virgules "...","...","...","..."

Ce champ contient la liste du ou des auteurs principaux du dictionnaire.

5.15 [altAuthors]

Format : Liste de textes entre guillemets séparés par des virgules "...","...","...","..."

Ce champ contient la liste du ou des auteurs accessoires du dictionnaire.

5.16 [contactAuthor]

Format : Texte (entre guillemets).

Ce champ contient des indications pour contacter le ou les auteurs : adresses de courrier ou courriel, téléphone, etc.

5.17 [shortAuthors]

Format : Texte (entre guillemets).

Ce champ contient une information abrégée sur le ou les auteurs. Celle-ci est destinée aux affichages nécessitant de la brièveté quand le contenu cumulé de la liste du champ *mainAuthors* risque d'être trop long.

Important : la formulation de ce champ est libre MAIS elle doit être *la plus courte possible*.

Exemples :

```
mainAuthors="Pierre Martin – 75200 Paris"
shortAuthors="P.Martin"
```

```
mainAuthors="Pierre Martin - Paris","André Lajoie - Marseille"
shortAuthors="Martin & Lajoie"
```

```
mainAuthors="Pierre Martin","André Lajoie","Marcel Legrand","Paul Dubois"
shortAuthors="P.Martin & al."
```

5.18 [dicStatus]

Format : Texte (entre guillemets).

Ce champ contient les informations de statut et de licence du dictionnaire.

En cas de référence à un type public de licence, il est souhaitable d'inclure l'url du texte complet de la licence sur le Web.

5.19 [showDicStatus]

Format : Booléen.

Ce champ indique au logiciel de gestion du dictionnaire si le statut du dictionnaire (contenu du champ [dicStatus]) doit s'afficher ou non dans l'interface du logiciel.

Les modalités et emplacements de cet affichage relèvent ensuite des choix de conception des logiciels.

5.20 [copyright]

Format : Texte (entre guillemets).

Ce champ contient une mention de copyright ou équivalent (référence aux droits d'auteur). Ce champ est destiné à contenir une mention courte s'affichant sur une ligne. Pour les indications détaillés concernant les auteurs, le statut et la licence ce sont les champs [...Authors] et [dicStatus] qui doivent être utilisés.

5.21 [creationDate]

Format : Texte (entre guillemets).

Ce champ contient la date de création initiale du dictionnaire, c'est-à-dire la date de sa première publication et non celle de la version courante.

Cette date doit être indiquée en format ISO : yyyy-mm-dd

Exemple :

```
creationDate="2008-06-09"          # 9 juin 2008
```

5.22 [versionDate]

Format : Texte (entre guillemets).

Ce champ contient la date de publication de la version courante du dictionnaire.

Cette date doit être indiquée en format ISO: yyyy-mm-dd

Exemple :

```
versionDate="2009-09-05"          # 5 septembre 2009
```

5.23 [localEditDate]

Format : Texte (entre guillemets).

Ce champ contient la date la plus récente à laquelle le dictionnaire a été édité (modifié) par l'utilisateur.

Cette date doit être indiquée en format ISO : yyyy-mm-dd

Les logiciels doivent mettre jour ce champ à chaque modification du dictionnaire par l'utilisateur, ce qui est beaucoup plus fiable que de se baser sur la date de modification du fichier renvoyée par le système de fichiers de la machine.

Exemple :

```
localEditDate="2009-09-05"        # 5 septembre 2009
```

5.24 [dicID]

Format : Texte (entre guillemets).

Ce champ contient l'identifiant du dictionnaire. Cet identifiant doit être *unique* et *définitif* dans un ensemble de dictionnaires donné.

Un tel identifiant n'a de sens qu'en relation avec une norme externe à la présente spécification.

Le format LING utilisant des wordIDs, il est donc possible de pointer à partir d'un dictionnaire sur un mot d'un autre dictionnaire LING en utilisant par exemple une syntaxe qualifiée de type "dicID.wordID". Le format LING incluant une table des wordIDs, extraire un mot d'un dictionnaire externe ne nécessite donc que le parcours de cette table et celle des blocs de cartographie pour accéder directement au mot pointé dans le fichier externe.

5.25 [dicVersionNumber]

Format : Texte (entre guillemets).

Ce champ contient le numéro de version du dictionnaire.

Attention, ce "numéro" n'est pas une valeur numérique mais une valeur textuelle.

5.26 [dicUrl]

Format : Texte (entre guillemets).

Ce champ contient une url permettant d'accéder au dictionnaire sur le Web, soit un lien de téléchargement direct soit, ce qui est préférable, une page permettant d'accéder secondairement au téléchargement du dictionnaire.

5.27 [verUrl]

Format : Texte (entre guillemets).

Ce champ contient une url représentant un lien de téléchargement d'un fichier texte quelconque contenant des informations sur la version du dictionnaire disponible en téléchargement en ligne.

Dans ce fichier texte, l'information du numéro de version devra se trouver isolée sur une ligne et formatée ainsi en utilisant l'identifiant du dictionnaire suivi de "_version" :

```
<dicID>_version=00.00.00
```

Si le dictionnaire n'a pas de dicID défini :

```
_version=00.00.00
```

Pour intégrer cette ligne à un fichier texte pour le Web (pur texte, html, javascript, php, etc.), il suffit de la placer dans un bloc de commentaires multilignes :

Exemples :

```
<html>
<--!
MonDicID_version=00.00.00
-->
<?php
/*
MonDicID_version=00.00.00
```

```
*/
?>
</html>
```

Voir également : Champ [dicID]

5.28 [dicInfo]

Format : Texte (entre guillemets).

Ce champ contient un texte d'information et de présentation du dictionnaire, destiné à l'utilisateur.

5.29 [showDicInfo]

Format : Booléen.

Ce champ indique au logiciel de gestion du dictionnaire si le texte de présentation du dictionnaire (contenu du champ [dicInfo]) doit s'afficher ou non dans l'interface du logiciel.

Les modalités et emplacements de cet affichage relèvent ensuite des choix de conception des logiciels.

5.30 [protected1] , [protected2]

Format : Texte (entre guillemets).

Ce champ contient le nom du protocole de protection/cryptage utilisé pour protéger le bloc des entrées (*protected1*) et le bloc des notices (*protected2*).

La présente spécification ne définit aucun protocole de cryptage précis lié au format LING. Ces protocoles relèvent du seul domaine des logiciels. Les seules obligations qui doivent être suivies par ceux-ci sont :

- La protection du dictionnaire ne doit pouvoir être appliquée ou levée que par voie d'importation textuelle (voir spécification PRELING) ce qui impose de posséder le document-source du dictionnaire pour en gérer la protection.
- Une suppression ou une modification manuelle de ce(s) champ(s) dans le fichier LING protégé ne doit pas pouvoir lever la protection une fois celle-ci appliquée.

5.31 [displayFontName1] , [displayFontName2]

Format : Texte (entre guillemets).

Ces champs contiennent des noms de polices d'affichage conseillées pour la langue source (*displayFontName1*) et la langue cible (*displayFontName2*).

Syntaxe : "<nom de la police>**?**<taille>**?**<attributs>"

Si la valeur est une chaîne vide ou si le champ est absent, la police correspondante sera déterminée par le logiciel.

La gestion de l'éventuelle priorité de ces polices sur les polices définies en interne dans les logiciels est de la responsabilité de ces derniers. La seule obligation est que si les polices désignées par ces champs sont absentes de la machine de l'utilisateur, cette absence ne doit générer aucune erreur dans les logiciels.

Exemples :

```
displayFontName1="Helvetica?9?normal"
displayFontName2="Times new Roman?10?bold"
```

5.32 [grammarEncoding1]

Format : Texte (entre guillemets).

Ce champ contient une indication sur la convention utilisée pour rédiger les données grammaticales associées aux entrées de la langue source : ordre, abréviations, conteneurs, etc.

Cette indication permet d'informer les logiciels quant au type de translittération et d'affichage à utiliser pour les données grammaticales associées aux entrées. Le contenu de ce champ n'a donc de sens que si les logiciels gère la norme de codage indiquée, sinon ils doivent l'ignorer.

5.33 [compatPlugins] , [noCompatPlugins] [usePlugins]

Format : Liste de textes entre guillemets séparés par des virgules "...","...","...","..."

Ces champs donnent aux logiciels des indications sur l'utilisation éventuelle de greffons (plugins en anglais) avec le dictionnaire.

Les champs [compatPlugins] et [noCompatPlugins] contiennent une liste de greffons compatibles ou incompatibles avec le dictionnaire. Le nom du logiciel concerné doit être associé au greffon par ":".

Le champ [usePlugins] contient une liste de greffons à charger ou activer, si possible, lors du chargement du dictionnaire. Cette liste est nécessairement une sous-liste de celle de [compatPlugins]

Exemples :

```
compatPlugins="PrgMachin:GrefTruc","PrgBidule:GrefChose"
nocompatPlugins="PrgMachin:GrefXXX","PrgBidule:GrefZZZ"
usePlugins="PrgMachin:GrefTruc"
```

5.34 [wordGroups]

Format : Liste de textes entre guillemets séparés par des virgules "...","...","...","..."

Ce champ contient les déclarations des identifiants de groupes de mots qui seront ensuite utilisés dans le sous-champ *wg* du champ des attributs des notices (cf.).

Un identifiant de groupe ne doit pas contenir d'espaces mais peut contenir des tirets et des symboles de soulignement.

Le libellé de l'identifiant de groupe peut être facultativement suivi d'un alias plus convivial en utilisant une barre verticale pour les séparer.

L'usage qui est fait de ces groupes de mots est du domaine des logiciels de gestion des dictionnaires.

Exemple :

```
wordGroups="arch|Architecture","mar|Terme de marine","inf|Informatique"
```

5.35 [biblio]

Format : Liste de textes entre guillemets séparés par des virgules "...","...","...","..."

Ce champ contient une liste de références bibliographiques pour le dictionnaire.

Le libellé de chaque référence peut être facultativement suivi d'un identifiant (quelconque mais *unique* dans le cadre de ce champ, le plus souvent un numéro) en utilisant une barre verticale pour les séparer. Cet identifiant permet de pointer sur une référence bibliographique à partir de la notice d'une entrée.

Exemple :

```
biblio="Machin A. (1893) - Dictionnaire Français-Suédois|1","Bidule H. (1932) - Structure du suédois|2","Trucmuche R. (1975) – Terminologie maritime scandinave|3"
```

5.36 [showBiblio]

Format : Booléen.

Ce champ indique au logiciel de gestion du dictionnaire si la bibliographie complète (le contenu du champ [biblio] épuré de ses identifiants) doit s'afficher ou non dans l'interface du logiciel.

Les modalités et emplacements de cet affichage relèvent ensuite des choix de conception des logiciels.

5.37 [extFieldCount]

Format : valeur numérique.

Ce champ contient le nombre de champs d'extension qui doit être ajouté aux notices. Si ce champ est absent ce nombre est de zéro.

Des champs d'extension peuvent être librement ajoutés aux notices, ces champs sont propres au dictionnaire ou au logiciel utilisé. Leur nombre doit être déclaré ici.

Les champs d'extension qui ne sont pas gérés par les logiciels doivent être ignorés.

Nb : ne pas confondre les champs d'extension des notices et les champs de propriétés additionnelles. Ces derniers n'ont besoin d'aucune déclaration préalable.

5.38 [extFieldList]

Format : Liste de textes entre guillemets séparés par des virgules "...","...","...","..."

Ce champ contient la liste facultative des noms conviviaux donnés aux champs d'extension des notices. Ces noms peuvent être éventuellement utilisés par les logiciels pour l'affichage mais sont surtout une information interne sur la nature des champs d'extension, qui améliore la lisibilité des propriétés du dictionnaire.

Si la liste est moins longue que le nombre de champs, les noms sont affectés dans l'ordre aux premiers champs. Si elle est plus longue, les noms excédentaires sont ignorés.

6. Le Bloc des entrées du dictionnaire

Ce bloc est constitué d'une suite d'entrées encodées en utf-8.

Les "entrées" sont les "mots-titres" du dictionnaire, généralement destinés à être affichés sous la forme d'une liste.

Dans le bloc, les entrées sont séparées par <00>. Il n'y a aucun marqueur de début ou de fin de bloc (pas de <00> en fin de bloc).

```
Mot d'index 0 <00>     # Libellé du premier mot
Mot d'index 1 <00>     # Libellé du deuxième mot
Mot d'index 2 <00>     # Libellé du troisième mot
..............<00>     # etc...
..............<00>
..............<00>
Mot d'index X          # Libellé du dernier mot
```

Dans le dictionnaire, à chaque entrée sont associés un wordID (facultatif) et une notice.

7. Le Bloc des wordIDs

7.1 A propos des wordIDs

On désigne par wordID un identifiant arbitraire rattaché à une entrée de dictionnaire.

Chaque wordID doit être **unique** dans le dictionnaire. Tout dictionnaire contenant des doublons de wordID doit être considéré comme invalide par les logiciels.

A chaque wordID doit correspondre *une* et *une seule* entrée mais toutes les entrées n'ont pas forcément un wordID associé.

Syntaxe d'un wordID :

- 1 caractère minimum, 8 caractères maximum.
- Uniquement des caractères alphanumériques 7-bits (ascii strict) minuscules.

- Les majuscules, les espaces et le caractère de soulignement sont interdits.

Hormis pour ce qui précède, le libellé des wordIDs est libre mais utiliser, quand c'est possible, les 3 premières lettres (en minuscules) de l'entrée suivies d'un numéro est une habitude commode, compacte et lisible.

A noter qu'utiliser systématiquement des wordIDs courts (moins de 8 caractères) ne change rien à la taille du fichier du dictionnaire, car un libellé de wordID y est toujours enregistré sur 8 octets quelle que soit sa longueur réelle.

7.2 Structure du bloc

A chaque wordID est associé l'index de l'entrée dans le bloc des entrées (ou son index dans le bloc des notices, car les deux sont identiques) et l'offset de l'entrée dans le bloc des entrées.

Le wordID est inscrit sur 8 caractères (8 octets) même si sa longueur réelle est inférieure, dans ce cas le nombre nécessaire d'espaces est ajouté à sa *gauche*.

L'index de l'entrée est codé sur 32 bits (4 octets), octet lourd à gauche.

L'offset de l'entrée dans le bloc des entrées est codé sur 32 bits (4 octets), octet lourd à gauche.

Le wordID, l'index et l'offset sont inscrits de manière contiguë, sans séparateur. Les informations liées à chaque wordID occupent donc 8 + 4 + 4 = 16 octets.

```
-----wID<idx><offset>        # 8 + 4 + 4 = 16 octets
-----wID<idx><offset>        # 16 octets
-----wID<idx><offset>        # ...
......................
......................
-----wID<idx><offset>
```

Attention ! L'offset des entrées dans le bloc des entrées est un offset *relatif* à ce bloc. L'offset réel dans le fichier sera donc :

offset réel = offset du bloc des entrées + offset indiqué dans le présent bloc.

7.3 Utilités potentielles de ce bloc

1) Accélérer le chargement du dictionnaire :

Les logiciels disposent ainsi d'une table des wordIDs directement utilisable sans avoir à parcourir toutes les notices pour bâtir à la volée cette table à chaque chargement du dictionnaire.

2) Faciliter les relations externes entre dictionnaires :

Pour pointer de manière externe sur un mot d'un dictionnaire LING, et l'extraire sans devoir charger le dictionnaire ou en parcourir séquentiellement tout le fichier, il faut :

- Charger les informations de navigation du bloc de cartographie globale (70 premiers octets du fichier)
- Se déplacer en tête du bloc des wordsIDs
- Rechercher le wordID dans ce bloc en le parcourant par plages de 16 octets. Une fois trouvé le wordID, on a donc l'index et l'offset de l'entrée.
- L'offset de l'entrée + l'offset du bloc des entrées donne l'offset du libellé de l'entrée dans le fichier.
- L'index de l'entrée donne les coordonnées des notices dans le fichier, en consultant pour cela la table du bloc de cartographie des notices.

L'extraction d'un élément de dictionnaire (entrée et notice associée) ne nécessite donc qu'une recherche séquentielle dans le bloc des wordIDs, une fois le wordID trouvé tout le reste de l'extraction se fait par adressage direct donc de manière rapide et économe en mémoire.

8. Le Bloc de cartographie des notices

Ce bloc de cartographie est une suite de nombres encodés sur 32 bits (4 octets), octet lourd à gauche.

Offset et taille sur 32 bits, soit 8 octets par entrée. Pour extraire les coordonnées de la notice d'index n, il faut donc lire 8 octets à partir de :

offset du présent bloc + (n x 8)

Attention ! : les offsets indiqués dans ce bloc sont les offsets RELATIFS au bloc des notices et non au fichier.

offset réel = offset du bloc des notices + offset relatif indiqué dans ce bloc

```
XXXX XXXX  # [Offset de la notice 1][Taille de la notice 1]
XXXX XXXX  # [Offset de la notice 2][Taille de la notice 2]
XXXX XXXX  # [Offset de la notice 3][Taille de la notice 3]
.... ....  # etc...
.... ....
.... ....
```

```
XXXX XXXX  # [Offset de la notice z][Taille de la notice z]
```

9. Le Bloc des notices

Ce bloc contient les notices rattachées aux entrées. Elles sont encodées en utf-8.

9.1 Structure du bloc

Dans ce bloc, chaque notice est un ensemble de champs textuels séparés par <00>. Il n'y a pas de marqueur de début ou de fin de bloc (pas de <00> à la fin).

La présence de chaque champ est obligatoire (mais le champ peut rester vide).

La longueur du contenu de chaque champ n'est pas limitée, hormis le champ du wordID de l'entrée puisqu'un wordID ne peut dépasser 8 caractères (cf.)

```
Entrée 1,  Chp[0]<00> # Traductions courtes      <(début de la notice 1)
Entrée 1,  Chp[1]<00> # Traductions longues et Infos diverses
Entrée 1,  Chp[2]<00> # wordID de l'entrée
Entrée 1,  Chp[3]<00> # wordIDs des racines (séparés par ";")
Entrée 1,  Chp[4]<00> # wordIDs des synonymes (séparés par ";")
Entrée 1,  Chp[5]<00> # wordIDs des "voir aussi" (séparés par ";")
Entrée 1,  Chp[6]<00> # Attributs divers de l'entrée (séparés par ";")
Entrée 1,  Chp[7]<00> # Phonétique de l'entrée
Entrée 1,  Chp[8]<00> # wordIDs des antonymes (séparés par ";")
Entrée 1,  Chp[9]<00> # Contenu du Champ étendu 1
Entrée 1, Chp[10]<00> # Contenu du Champ étendu 2
     ...         <00>
     ...         <00>
Entrée 1, Chp[x]      # Contenu du Champ étendu x  <(fin de la notice 1)
Entrée 2, Chp[0]<00>  # Traductions courtes      <(début de la notice 2)
Entrée 2, Chp[1]<00>  # Traductions longues et Infos diverses
     ...         <00>  # ...
     ...         <00>
     ...         <00>
     ...               # ...                 <(fin de la dernière notice)
```

9.2 Balisage interne des données

Les champs contenant des données textuelles destinées à l'affichage (traductions et infos diverses) peuvent contenir des balises.

La présente spécification définit une liste de balises standards qui devraient être prise en charge par les logiciels. Néanmoins, toutes les balises sont autorisées (en particulier le balisage de type XDXF, cf.). Les logiciels ne reconnaissant pas une balise doivent simplement la supprimer de l'affichage tout en affichant le contenu balisé brut.

9.2.1 Balises de formatage visuel

Les balises suivantes sont autorisées dans les données

** **	: saut de ligne.
...	: texte en gras.
...	: texte en italique.
<u>...</u>	: texte souligné.
...	: diminuer la taille du texte.
<big>...</big>	: augmenter la taille du texte.

9.2.2 Balises sémantiques

Nb : leurs libellés sont repris du format XDXF.

<etym>...</etym>

Conteneur d'informations sur l'étymologie de l'entrée.

Si l'entrée dispose de racines qui sont présentes dans le dictionnaire, il est préférable d'utiliser le champ des wordIDs des racines (voir plus bas) en remplacement ou en complément de ce conteneur.

<synonym>...</synonym>

Conteneur des synonymes de l'entrée.

Si l'entrée dispose de synonymes présents dans le dictionnaire, il est préférable d'utiliser le champ des wordIDs des synonymes (voir plus bas) en remplacement ou en complément de ce conteneur.

Attention ! Les deux types d'indication de la synonymie – conteneur <synonym> ou champ des wordIDS des synonymes – *ne sont pas interchangeables*. Dans le conteneur, les synonymes sont indiqués *en clair*, c'est du texte littéral indépendant de l'entrée qu'il représente, alors que dans le champ des wordIDs les synonymes sont *référencés* par leur

identifiant, ce sont donc des liens insensibles aux possibles modifications du libellé des entrées vers lesquelles ils pointent. Il faut donc privilégier, si possible, l'usage des wordIDS, l'intérêt du conteneur <synonym> étant soit de stocker des synonymes absents du dictionnaire, soit de recevoir des libellés de synonymes lors d'une conversion de fichier d'un format ne gérant pas les wordIDs, soit de contenir des informations annexes à la liste des wordIDs.

<antonym>...</antonym>

Conteneur des antonymes de l'entrée.

Toutes les remarques à propos du conteneur des synonymes s'appliquent aux antonymes.

9.2.3 Autres balises

La gestion d'autres balises que celles définies ci-dessus est du domaine non de la présente spécification mais de celui des logiciels. On pourrait citer en particulier la gestion des balises XDXF de couleurs <c c=#xxxxxx>...</c>, etc.

9.3 Champ "Traductions courtes"

Ce champ contient des traductions directes ou des définitions courtes des entrées du dictionnaire.

Son contenu doit être rédigé de la manière la plus concise possible. Les traductions ou définitions multiples sont séparées par ";" (point-virgule).

C'est le contenu de ce champ qui doit être utilisé par les logiciels pour les inversions automatisées du dictionnaire suivant le schéma suivant :

Mot A = Mot B1 ; Mot B2

Mot C = Mot D ; Mot B1

>>>

Mot B1 = Mot A ; Mot C

Mot B2 = Mot A

Mot D = Mot C

9.4 Champ "Traductions longues et infos diverses"

Ce champ contient les commentaires longs et "littéraires", les infos diverses sur les entrées, rédigées sans aucune contrainte de place.

Seront également placées dans ce champ toutes les informations concernant l'étymologie, les synonymes, les antonymes et les renvois lorsque ceux-ci ne peuvent être pris en charge par des liens directs vers des wordIDS (voir : *Champs des wordIDs*), par exemple pour citer un synonyme qui est absent en tant qu'entrée du dictionnaire, ou pour apporter des informations complémentaires concernant un renvoi à l'aide d'un wordID. Il est souhaitable d'utiliser des balises sémantiques pour délimiter ce type d' élément (voir plus haut : *Balisage interne des données*) plutôt que d'inscrire des données brutes.

9.5 Champs des wordIDs des racines/synonymes/"voir aussi"/antonymes

Ces champs définissent les relations entre les entrées. Ils contiennent les wordIDs des entrées vers lesquelles pointent l'entrée concernée.

Les wordIDs y sont séparés par des ; (point-virgule) et leur nombre par champ est sans limitation.

Les logiciels doivent gérer la possibilité qu'un wordID pointe dans le vide (lien brisé).

9.6 Champ des attributs

Ce champ contient des sous-champs de nature diverse mais dont le contenu est en principe *non destiné à être directement affiché* : codes de classification, codes de contrôle, commutateurs, symboles, commandes, indicateurs et attributs divers, etc.

Les sous-champs contenant les attributs sont séparés entre eux par ";" (le point-virgule).

Les attributs sont facultatifs et leur nombre est libre.

Un attribut peut avoir une valeur ou non, dans ce dernier cas c'est alors un simple indicateur. Si l'attribut a une valeur, elle est introduite par "="

```
nomAttribut=valeurAttribut
```

La présente spécification définit une liste limitative d'attributs standards, mais il est possible d'ajouter un nombre illimité d'attributs non standards.

La casse (majuscule/minuscule) des noms des attributs est signifiante.

9.6.1 Attributs standards :

wg : groupes de mots

Liste du ou des identifiants des groupes de mot auxquels se rattache l'entrée de dictionnaire, par référence aux groupes définis dans le bloc des propriétés du dictionnaire.

En cas de groupes multiples pour une même entrée, l'ordre des groupes n'est pas indifférent, les groupes doivent être inscrits dans l'ordre de pertinence décroissante, le premier groupe étant donc le plus pertinent.

Syntaxe : la liste est introduite par "=" et les IDs des groupes de mots sont séparés par "," (la virgule).

(Rappel : un identifiant de groupe ne doit pas contenir d'espaces mais peut contenir des tirets et des symboles de soulignement. Un alias convivial peut lui être associé dans le champ de propriété [wordGroups] du dictionnaire)

Exemple :

```
wg=IDgroupe1,Idgroupe2,Idgroupe3
```

e : entrée éditée.

Cet indicateur ne prend aucune valeur. Sa présence indique que l'entrée de dictionnaire a été modifiée par l'utilisateur ou le logiciel.

n : entrée ajoutée.

Cet indicateur ne prend aucune valeur. Sa présence indique que l'entrée de dictionnaire a été ajoutée par l'utilisateur ou le logiciel.

r : non inversion.

Cet indicateur ne prend aucune valeur. Sa présence indique que l'entrée de dictionnaire *ne doit pas* être incluse dans un dictionnaire inverse construit par un processus automatique.

9.6.2 Attributs non standards

Ces attributs sont propres à l'utilisateur ou au logiciel.

Un logiciel ne reconnaissant pas un attribut doit l'ignorer mais le respecter (c'est-à-dire le conserver tel quel, sauf intervention expresse de l'utilisateur) lors des processus d'édition de l'entrée concernée.

Le nombre des attributs non standards est libre ainsi que leur contenu.

Syntaxe :

La syntaxe de la valeur des attributs non standards est libre avec la limitation suivante : les valeurs *ne doivent contenir ni zéro binaire ni point-virgule*.

9.7 Le champ "Phonétique de l'entrée"

Ce champ contient la représentation textuelle de la phonétique du mot en entrée.

Pour éviter des problèmes de polices et d'affichage avec l'alphabet phonétique international, il est fortement conseillé d'utiliser une codification ASCII de la phonétique telle que SAMPA ou X-SAMPA.

10. Les Blocs des images

Chacun de ces deux blocs *facultatifs* contient l'image servant d'icône à la langue concernée

Un bloc est constitué de la chaîne base64 de l'image précédée du nom du type de l'image (format graphique) et en est séparée par <00>. Il n'y a pas de marqueurs de début ou de fin de bloc.

Le codage base64 est un standard de représentation en ASCII de données binaires.

Le format graphique de ces images est libre (gif, png, jpeg, bmp, tiff, ...) mais les formats les plus adaptés à cet usage sont le gif et le png.

Les logiciels incapables de traiter le format graphique présent dans le fichier s'abstiendront de l'affichage de l'image mais sans perte du code de l'image dans les traitements ultérieurs du fichier.

```
<type image 1><00><code de l'image 1 en base64>
```

```
<type image 2><00><code de l'image 2 en base64>
```

Pourquoi coder les images en format texte dans un fichier binaire ? Cela peut sembler un inutile gaspillage d'espace (le code base64 est environ 30% plus gros que le code binaire correspondant). La raison en est de faciliter les conversions "en aller-et-retour" entre le format Ling et son format texte passerelle Preling (cf.), le code des images restant ainsi le même dans les deux formats.

11. Mémento du format LING

Toutes les données textuelles sont encodées en Utf-8

Bloc identifiant et cartographie (70 octets)

Cartographie : nombres sur 32 bits (4 octets), octet lourd à gauche.

```
%ling/01.01.00   # octets 1-14  : Identifiant de fichier et version LING
XX XX            # octets 15-18 : Offset du bloc des Propriétés
XX XX            # octets 19-22 : Taille du bloc des Propriétés
XX XX            # octets 23-26 : Offset du bloc des Entrées
XX XX            # octets 27-30 : Taille du bloc des Entrées
XX XX            # octets 31-34 : Offset du bloc des wordIDs
XX XX            # octets 35-38 : Taille du bloc des wordIDs
XX XX            # octets 39-42 : Offset du bloc Cartographie des Notices
XX XX            # octets 43-46 : Taille du bloc Cartographie des Notices
XX XX            # octets 47-50 : Offset du bloc des Notices
XX XX            # octets 51-54 : Taille du bloc des Notices
XX XX            # octets 55-58 : Offset du bloc Image1 (00 00 si absent)
XX XX            # octets 59-62 : Taille du bloc Image1 (00 00 si absent)
XX XX            # octets 63-66 : Offset du bloc Image2 (00 00 si absent)
XX XX            # octets 67-70 : Taille du bloc Image2 (00 00 si absent)
```

Bloc des Propriétés du dictionnaire

les champs sont séparés par <00>

```
minCompatVersion="..."       # (Txt) Compatibilité LING minimale
maxCompatVersion="..."       # (Txt) Compatibilité LING maximale
dicName="..."                # (Txt) Nom convivial du dictionnaire
langName1="..."              # (Txt) Nom convivial de la langue 1
langName2="..."              # (Txt) Nom convivial de la langue 2
langIso1="..."               # (Txt) Code ISO 639-2 de la langue 1
langIso2="..."               # (Txt) Code ISO 639-2 de la langue 2
langNameUser="..."           # (Txt) Nom de la langue de l'utilisateur
langIsoUser="..."            # (Txt) Code ISO 639-2 langue utilisateur
langFamily1="..."            # (Txt) Famille linguistique de la langue 1
langFamily2="..."            # (Txt) Famille linguistique de la langue 2
isReverseDic= True|False     # (Bool) est un dictionnaire inverse auto
doReverseDic= True|False     # (Bool) est un dico conçu pour être inversé
reverseDicFileName="..."     # (Txt) Nom fichier du dico inverse/orig.
reverseDicName="..."         # (Txt) Nom convivial du dico inverse/orig.
sortEquPatterns="...","..." # (Txt-lst) Motifs d'équivalence de tri
```

```
sortEquPatternsRev="..","..."# (Txt-lst) Equ. de tri (dico inverse)
wordcount=...                # (Num) Nombre d'entrées du dictionnaire
mainAuthors="...","..."      # (Txt-lst) Auteur(s) principaux du dico
altAuthors="...","..."       # (Txt-lst) Auteur(s) accessoires
contactAuthor="..."          # (Txt) Contact des auteurs
shortAuthors="..."           # (Txt) auteurs, formulation courte
dicStatus="..."              # (Txt)Statut et licence du dictionnaire
showDicStatus= True|False    # (Bool) Afficher le statut du dictionnaire
copyright="..."              # (Txt) Mention de copyright ou équivalent
creationDate="..."           # (Txt) Date de création du dictionnaire
versionDate="..."            # (Txt) Date de la présente version du dico
localEditDate="..."          # (Txt) Date d'édition locale
dicID="..."                  # (Txt) Identifiant du dictionnaire
dicVersionNumber="..."       # (Txt) N° de version du dictionnaire
dicUrl="..."                 # (Txt) URL du dictionnaire sur le Web
verUrl="..."                 # (Txt) URL d'un éventuel fichier annexe
dicInfo="..."                # (Txt) Texte de présentation du dico
showDicInfo= True|False      # (Bool) Afficher les infos
protected1="..."             # (Txt) Protocole de protection des entrées
protected2="..."             # (Txt) Protocole de protection des notices
displayFontName1="..."       # (Txt) Police d'affichage de la langue 1
displayFontName2="..."       # (Txt) Police d'affichage de la langue 2
grammarEncoding1="..."       # (Txt) Norme codage grammatical langue 1
compatPlugins="...","..."    # (Txt-lst) Greffons compatibles
noCompatPlugins="...","..."  # (Txt-lst) Greffons incompatibles
usePlugins="...","..."       # (Txt-lst) Greffons à associer au dico
wordGroups="...","..."       # (Txt-lst) IDs des groupes de mots du dico
biblio="...","..."           # (Txt-lst) Références biblio du dico
showBiblio=True|False        # (Bool) Afficher les références biblio
extFieldCount=...            # (Num) Nbre champs d'extension des notices
extFieldList="...","..."     # (Txt-lst) Noms des champs d'extension
<autre_champ1>="..."         # (Txt, Bool, Num. ou Txt-lst)
<autre_champ2>="..."         # (Txt, Bool, Num. ou Txt-lst)
```

Bloc des Entrées

```
Mot d'index 0 <00>     # Libellé du premier mot
Mot d'index 1 <00>     # Libellé du deuxième mot
Mot d'index 2 <00>     # Libellé du troisième mot
..............<00>     # etc...
..............<00>
..............<00>
Mot d'index X          # Libellé du dernier mot
```

Bloc des wordIDs

Chaque wordID doit être unique dans le dictionnaire.

- wordID : 8 caractères maxi, 7-bits (ascii strict) en minuscules, '_' (souligné) interdit.

- index de l'entrée : sur 32 bits (4 octets), octet lourd à gauche.

- offset *relatif* de l'entrée dans le bloc des entrées : sur 32 bits (4 octets), octet lourd à gauche.

```
-----wID<idx><offset>       # 8 + 4 + 4 = 16 octets
-----wID<idx><offset>       # 16 octets
-----wID<idx><offset>       # ...
.....................
.....................
-----wID<idx><offset>
```

Bloc de cartographie des notices

Cartographie : nombres sur 32 bits, soit 8 octets par notice.

Offsets *relatifs* au bloc des données et non au fichier.

```
XXXX XXXX  # [Offset de la notice 1][Taille de la notice 1]
XXXX XXXX  # [Offset de la notice 2][Taille de la notice 2]
XXXX XXXX  # [Offset de la notice 3][Taille de la notice 3]
.... ....  # etc...
.... ....
.... ....
XXXX XXXX  # [Offset de la notice z][Taille de la notice z]
```

Bloc des notices

- Les champs internes à la notice sont séparés par <00>

- les sous-champs du champ des attributs sont facultatifs et libres

- attributs standards:

wg=... , ... (groupes de mots)

e (flag d'entrée éditée par l'utilisateur)

n (flag d'entrée ajoutée par l'utilisateur)

r (flag de non inversion)

```
Entrée 1,  Chp[0]<00> # Traductions courtes      <(début de la notice 1)
Entrée 1,  Chp[1]<00> # Traductions longues et Infos diverses
Entrée 1,  Chp[2]<00> # wordID de l'entrée
Entrée 1,  Chp[3]<00> # wordIDs des racines (séparés par ";")
Entrée 1,  Chp[4]<00> # wordIDs des synonymes (séparés par ";")
Entrée 1,  Chp[5]<00> # wordIDs des "voir aussi" (séparés par ";")
Entrée 1,  Chp[6]<00> # Attributs divers de l'entrée (séparés par ";")
```

```
Entrée 1,  Chp[7]<00>  # Phonétique de l'entrée
Entrée 1,  Chp[8]<00>  # wordIDs des antonymes (séparés par ";")
Entrée 1,  Chp[9]<00>  # Contenu du Champ étendu 1
Entrée 1, Chp[10]<00>  # Contenu du Champ étendu 2
     ...         <00>
     ...         <00>
Entrée 1, Chp[x]       # Contenu du Champ étendu x  <(fin de la notice 1)
Entrée 2, Chp[0]<00>   # Traductions courtes      <(début de la notice 2)
Entrée 2, Chp[1]<00>   # Traductions longues et Infos diverses
     ...         <00>  # ...
     ...         <00>
     ...         <00>
     ...               # ...                 <(fin de la dernière notice)
```

Blocs Image

Images symboliques de la langue 1 et 2 (drapeau, etc.)

Deux blocs facultatifs

```
<type image 1><00><code de l'image 1 en base64>

<type image 2><00><code de l'image 2 en base64>
```

Blocs non standards

Facultatifs, structure libre, nombre illimité

Le format passerelle PRELING

IMPORTANT : Les spécifications qui suivent n'ont de sens qu'en relation avec celle du format LING.

1. Généralités

Le format PRELING est un format passerelle de sauvegarde et d'échange (import/export) vers et à partir du format binaire LING.

Le format PRELING est un **pur format texte** donc éditable dans tout éditeur de texte (l'usage d'un tableur ou d'un tableau de traitement de texte facilite le travail).

Ce format est basé sur deux postulats de départ :

1) Un simple fichier CSV à deux champs est un fichier PRELING valide. Tous les ajouts à ce format minimaliste sont donc facultatifs.

2) TOUTES les potentialités du format LING sont accessibles par le format PRELING

Usages du format PRELING :

- Format intermédiaire permettant d'éditer *manuellement* un dictionnaire au format binaire LING.

- Format d'importation d'un document texte vers le format LING

- Format d'exportation en mode texte d'un dictionnaire LING sans perte d'information.

- Format d'édition modulaire d'un dictionnaire (voir plus bas "_include").

Encodage du fichier :

Le fichier PRELING peut utiliser un encodage de caractère quelconque, (l'encodage peut être précisé dans la première ligne du fichier). Pour rédiger un dictionnaire mêlant de nombreux caractères différents (caractères latins et caractères cyrilliques par exemple), il est impossible d'utiliser un encodage 8-bits et il est alors nécessaire d'utiliser un encodage basé sur UNICODE tel que l'Utf-8.

2. Structure d'un fichier PRELING :

2.1 Première ligne du fichier

Ligne facultative.

Elle se compose d'un identifiant de type de fichier (%preling), du nom de l'encodage utilisé pour le fichier et du séparateur des données.

Le séparateur des données est indiqué de manière littérale sauf pour le caractère de tabulation qui est remplacé par le symbole {tab}. Ce caractère est le séparateur par défaut si aucun séparateur n'est précisé.

Exemples :

```
%preling/utf-8/{tab}
```

```
%preling/utf-8/%
```

```
%preling/utf-8/===
```

A propos du séparateur des données voir plus bas : *Lignes de données*.

2.2 Lignes de commentaires

Les lignes commençant par "_" sont des commentaires

Les lignes vides sont autorisées.

Commentaires et lignes vides devront être négligées par le convertisseur Preling>Ling.

On peut placer des lignes de commentaires n'importe où sauf à l'intérieur d'un bloc image (voir plus bas).

2.3 Lignes d'include

La commande "include" est placée dans un faux commentaire : _include.

Cette commande permet d'inclure le contenu d'un fichier texte externe comme s'il faisait partie intégrante du document PRELING.

L'inclusion est effectuée à l'endroit précis de la commande _include dans le document-maître. Le contenu du fichier à inclure continue donc de manière séquentielle ce qui précède l'emplacement de la commande _include et il se continue par ce qui suit l'emplacement de cette commande.

Exemple :

```
_ La ligne suivante inclut le contenu du fichier xxx.txt
_include xxx.txt
```

Les lignes de commandes _include sont utilisables n'importe où, au milieu des lignes de propriétés ou des lignes de données, sauf dans un bloc image (voir plus bas).

Le contenu des fichiers à inclure est libre : commentaires, lignes de propriétés, lignes de données, blocs images et le mélange de tout ceci.

Attention, à utiliser le *même* séparateur des données dans le document-maître et les fichiers à inclure.

La résolution des _include (remplacement des commandes par le contenu des fichiers vers lesquels elles pointent) sera effectuée par les logiciels au moment de l'importation du document PRELING et de sa conversion en dictionnaire LING.

Intérêt de l'utilisation d'_include :

L'utilisation de commandes _include permet de rédiger des dictionnaires de façon totalement modulaire. A l'extrême, un document-maître ne contiendra plus qu'une succession d'_include.

Exemple :

```
%preling/utf-8/{tab}
_include properties1.txt
_include properties2.txt
_include verbes.txt
_include noms.txt
_include adjectifs.txt
_include prepositions.txt
_include adverbes.txt
_include divers.txt
_include images.txt
```

La construction modulaire présente deux avantages :

- Un gros dictionnaire est ainsi fractionné en plusieurs petits documents, plus faciles à rédiger et à ordonner qu'un énorme document unique.

- Un dictionnaire devient un assemblage de briques thématiques *réutilisables*. Ainsi, la réalisation de multiples dictionnaires spécialisés peut reprendre, par exemple, des modules tels que "adverbes" et "prépositions" issus d'un dictionnaire général, d'où gain de temps, moins de risque d'oublis et fiabilité supérieure. En effet, l'utilisation d'_include n'est pas comparable à un simple copier-coller de bouts de dictionnaires déjà rédigés, car le fait de corriger ou mettre à jour l'un des modules met à jour automatiquement, lors de leur compilation en dictionnaires LING, *tous* les dictionnaires utilisant ce module !

2.4 Lignes de propriété

Lignes facultatives.

Propriétés standards :

Les lignes de propriétés standards reprennent les noms des champs LING standards (cf. plus haut) en les faisant précéder de "::" et la valeur du champ est introduite par "=".

Contrairement au format LING, les guillemets encadrant les valeurs textuelles isolées sont facultatives, mais elles restent obligatoires pour les listes de valeurs textuelles (les éléments individuels de ces listes sont séparés par ",") Les éventuels guillemets doubles internes au texte des propriétés doivent être inscrits sous le forme de l'entité Html `&quot;`

```
::dicName=Français - Suédois
::langName1="Français"
::langName2=Suédois
::mainAuthors="Moi","Lui","Un autre aussi"
:: ...
:: ...
```

Voir la spécification LING pour plus d'informations sur la nature et la syntaxe des champs de propriété.

Propriétés additionnelles :

Les propriétés additionnelles (propriétés non standards) sont autorisées.

Le nom d'une propriété additionnelle doit être préfixé par "::x_ling_" et la valeur du champ est introduite par "=".

L'interprétation du type d'une propriété additionnelle se fera ainsi :

- La valeur débute par une lettre ou des guillemets : texte.
- La valeur débute par un chiffre : numérique.
- La valeur vaut True ou False : booléen.

```
::x_ling_NomDuChamp1=valeur1
::x_ling_NomDuChamp2="valeur2a","valeur2b","valeur2c"
::x_ling_nomDuChamp3=valeur3
::x_ling_ ...
::x_ling_ ...
```

2.5 Lignes de données

Chaque ligne correspond à un élément de dictionnaire : le mot d'entrée et la notice associée. Une ligne peut donc être très longue.

Chaque ligne est segmentée en champs par un caractère séparateur qui est par défaut le caractère de tabulation.

Seuls deux champs sont obligatoires et *le nombre de champs peut varier entre les lignes d'un même fichier PRELING*.

A l'intérieur d'un champ, les données sont segmentées (si besoin) par ";" (point-virgule)

Les sauts de lignes sont interdits dans une ligne de donnée. Pour placer un saut de ligne dans un champ textuel, il faut le remplacer par la balise <br>.

Les balises suivantes sont utilisables dans les champs des notices qui sont destinés à être affichés :

-
 : saut de ligne. [OBLIGATOIRE]
- ... : texte en gras.
- ... : texte en italique.
- <u>...</u> : texte souligné.
- <small>...</small> : diminuer la taille du texte.
- <big>...</big> : augmenter la taille du texte.
- <etym>...</etym> : conteneur de l'étymologie.
- <synonym>...</synonym> : conteneur des synonymes
- <antonym>...</antonym> : conteneur des antonymes

Les balises sont écrites en *minuscules*.

Il ne doit pas y avoir de balises dans le champ du mot d'entrée de dictionnaire.

Voir plus haut dans *Bloc des notices du format Ling* pour plus d'information sur la fonction de ces balises.

Le séparateur des données :

Le séparateur peut être défini dans la première ligne du fichier (voir plus haut). S'il n'est pas explicitement défini, c'est le caractère de tabulation qui est considéré par défaut comme séparateur des données.

Le séparateur peut être constitué d'un ou plusieurs caractères en pur ascii (7bits).

Le caractère ou le motif de caractères doit être choisi de telle manière qu'il soit *absent des données*.

Les champs :

Champ [0] : Texte de l'entrée de dictionnaire

Champ [1] : Traductions/définitions courtes (séparées par ";")

Champ [2] : Infos et traductions/définitions détaillées

Champ [3] : wordID de l'entrée

Champ [4] : wordIDs des racines (séparés par ";")

Champ [5] : wordIDs des synonymes (séparés par ";")

Champ [6] : wordIDs des "voir aussi" (séparés par ";")

Champ [7] : Attributs divers (séparés par ";")

Champ [8] : Phonétique de l'entrée

Champ [9] : wordIDs des antonymes (séparés par ";")

Champ [10]: Champ étendu n°1

...

Champ [xx]: Dernier champ étendu

Voir la spécification LING pour plus d'information sur les champs des notices.

```
_ Une ligne de données :
Chp[0]<sep>Chp[1]<sep>Chp[2]<sep>Chp[3]<sep>Chp[4]<sep>Chp[5]<sep>Chp[6]
<sep>Chp[7]<sep>Chp[8]<sep> ... <sep>Chp[x]


_ Une autre ligne de données :
Chp[0]<sep>Chp[1]<sep>Chp[2]<sep>Chp[3]<sep>Chp[4]<sep>Chp[5]<sep>Chp[6]
<sep>Chp[7]<sep>Chp[8]<sep> ... <sep>Chp[x]


_ Une ligne de données incomplète (mais valide) :
Chp[0]<sep>Chp[1]<sep>Chp[2]<sep>Chp[3]
```

```
...
...
...
```

2.6 Les blocs image

Deux blocs de code encodé en *base64* (format texte) représentant les icônes du dictionnaire.

Après les deux mots-clés **imgxbegin le nom du format graphique de l'image est introduit par ":". Si ce format n'est pas précisé, le code de l'image est considéré comme étant au format Gif.

Un bloc image occupe trois lignes : une ligne *imgbegin* une ligne *base64* et une ligne *imgend*.

```
**img1begin:<format de l'image>
xxxxxxxxxxxxxxx(code de l'image 1 en base64)xxxxxxxxxxxxxxxx
**img1end

**img2begin:<format de l'image>
xxxxxxxxxxxxxxx(code del'image 2 en base64)xxxxxxxxxxxxxxxx
**img2end
```

Le codage *base64* est un standard de représentation en pur ASCII (7-bits) de données binaires. De nombreux petits utilitaires permettent la conversion binaire > base64.

Voici, comme exemple, le bloc image d'un petit drapeau français au format gif :

```
**img1begin:gif
R0lGODlhGAASALMAAAAA//8AAP//////////////////////////////////////
////////ywAAAAAGAASAAAEPBDIKYW9NuitKcUYx3kTeIkbWZkC2qls66pA7Aa0PcPsnfc7kg
71M/l4xqBnKCqCjkIgEflUfqRNasgVAQA7
**img1end
```

3. La conversion PRELING >LING

La conversion d'un fichier PRELING en fichier LING doit reprendre l'intégralité des données présentes, y compris les propriétés additionnelles (propriétés non standards).

Le fait que le fichier LING généré par la conversion soit trié alphabétiquement ou reprenne l'ordre des entrées du fichier PRELING est de la responsabilité des logiciels et non du domaine de la présente spécification.

_____