
Paramétrer et migrer ses données vers Koha 2.2

Paul POULAIN <paul AT koha-fr.org>

2005-01-06

Revision 2.2.0

Revision History

2005-01-06

pp

Version initiale

Table of Contents

1. Introduction	1
1.1. Utilisation de ce document	1
1.2. Généralités	2
1.3. Pré-requis	2
2. Première itération	3
2.1. dumpmarc.pl	3
2.2. Paramétrage UNIMARC	3
2.3. Premier import	6
2.4. Premiers résultats	6
2.5. Amélioration de la grille de saisie	8
2.6. Cas de la migration d'un fichier non iso2709	14

1. Introduction

1.1. Utilisation de ce document

1.1.1. Copyright and License

Copyright 2004 Paul Poulain <paul.poulain AT free.fr>

This document is related to Koha and is licensed to you under the GNU General Public License version 2 or later (<http://www.gnu.org/licenses/gpl.html>).

Koha-related documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies.

You may create a derivative work and distribute it provided that you:

1. License the derivative work with this same license, or the Linux Documentation Project License (<http://www.tldp.org/COPYRIGHT.html>). Include a copyright notice and at least a pointer to the license used.
2. Give due credit to previous authors and major contributors.

Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.

1.1.2. Disclaimer

No liability for the contents of this document can be accepted. Use the concepts, examples and information at your own risk. There may be errors and inaccuracies, that could be damaging to your system. Proceed with caution, and although this is highly unlikely, the author(s) do not take any responsibility.

All copyrights are held by their by their respective owners, unless specifically noted otherwise. Use of a term in this document should not be regarded as affecting the validity of any trademark or service mark. Naming of particular products or brands should not be seen as endorsements.

1.2. Généralités

La migration de ses données se fait de manière itérative :

- On examine les données à migrer (on suppose un fichier iso2709)
- On fait une partie du paramétrage
- On importe les notices et on regarde
- On affine le paramétrage des notices bibliographiques
- On réimporte les notices...
- Lorsque l'import des notices bibliographiques est terminé et validé, on peut passer à la migration des notices d'autorité, qui se fait également de manière itérative.

1.3. Pré-requis

La migration des données suppose des compétences métiers ET informatiques. Ne tentez pas de vous lancer si vous n'avez pas ces deux compétences. Si vous êtes informaticien vous risqueriez de décevoir vos utilisateurs. Si vous êtes bibliothécaire, ce qui suit va vous paraître totalement obscur.

Si vous ne pouvez pas réunir les deux compétences, il serait plus prudent d'envisager une aide externe...

Pré-requis techniques :

- Installation de Koha : on suppose que Koha est installé. \$KOHA est le répertoire dans lequel se trouve l'interface bibliothécaire.

```
ls -l $KOHA
```

doit renvoyer : cgi-bin htdocs modules scripts

- Le fichier de configuration de Koha se trouve dans /etc/koha.conf
- Toutes les manipulations décrites ci-après sont effectuées sur une console (shell) sur le serveur.
- se placer dans un shell (ligne de commande) et vérifier que Perl peut trouver vos modules :

```
export PERL5LIB=$KOHA/modules
```

2. Première itération

2.1. dumpmarc.pl

Se placer dans le répertoire ou le fichier des notices à migrer se trouve. Nous allons supposer qu'il s'appelle notices.iso.

Pour examiner son contenu, le script dumpmarc.pl vous sera utile :

```
$KOHA/scripts/misc/dumpmarc.pl -f notices.iso
```

L'ensemble des notices est affiché sous une forme lisible par un humain normalement constitué, ce qui n'est pas le cas d'une notice "iso2709" brute.

Vous pouvez stopper l'affichage par CTRL-C si vous avez beaucoup de notices !

Examinez les notices à l'écran, et notez les champs qui sont fréquemment utilisés. Il y aura probablement (en UNIMARC) :

- 200, avec titre et auteurs
- 205, pour les mentions d'édition
- 210 pour l'adresse bibliographique
- 215 pour la collation
- 010 pour l'ISBN et 011 pour l'ISSN
- 600 à 699 pour les autorités matières
- 700 à 799 pour les autorités auteurs

Mais ce document n'est pas un cours UNIMARC !

Vous devez également chercher les notices d'exemplaire. Ces dernières sont généralement dans le champ 995 si vos notices respectent bien la recommandation 995.

2.2. Paramétrage UNIMARC

Lorsque vous avez installé Koha, vous avez du choisir et donc importer tous les champs/sous-champs UNIMARC. Vous pouvez donc aller dans

```
Koha >> Paramètres >> Grilles de catalogage
```

Là, vous pouvez modifier la grille pour l'adapter à vos besoins et à ce que vous avez vu dans le paragraphe précédent.

2.2.1. Notion essentielle : base MARC / base non-MARC

L'une des contraintes majeures de l'équipe de développement de Koha est de fournir une application qui puisse être "multi-MARC".

Or, si toutes les déclinaisons MARC ont la même forme (l'iso2709), la signification de chaque champ/

sous champ diffère totalement d'une déclinaison à une autre.

Par exemple, en UNIMARC, le champ 200\$a contient le titre.

En MARC21, le titre est dans la zone 245\$a.

Le 245\$a en UNIMARC contient ... rien (la zone n'existe pas !)

Bref, il n'est pas possible de manière simple de gérer ces différences.

L'équipe de développement de Koha a donc choisi une méthode pour contourner ce problème, sans gréver les performances : tout est stocké en double

Note

ceux que ce stockage en double génèreront pourront toujours se consoler en se disant que le coût du Mo de disque est ridicule en regard du gain de performance obtenu en recherche ! D'ailleurs, sans rentrer dans les détails, toujours pour des questions de performance, les données sont en fait stockées en triple !

: sous la forme "MARC" (200\$a, qui ignore le sens du champ) et sous la forme "décodée" (titre qui ignore la position MARC).

Une table de paramétrage est mise en place lors de l'installation. Cette table contient non seulement les libellés de tous les champs et sous-champs UNIMARC (ou MARC21), mais également les liens entre les deux bases.

L'application, lors de l'ajout ou la modification des notices, se charge de stocker l'information deux fois.

Cela augmente fortement la complexité des opérations d'ajout/modification des notices. Mais simplifie d'autant les opérations de recherche et de lecture. Comme dans un SIGB 90% des accès sont des recherches et moins de 10% des ajouts/modifications, la rapidité de l'application s'en trouve renforcée.

Par exemple, lors de l'affichage d'une liste de résultats à une requête, on récupère le "title" directement. Si on devait passer par la notice unimarc, on devrait récupérer toute la notice pour en extraire le seul titre !

2.2.2. Paramétrage de la grille de catalogage

Notez que Koha sait gérer différentes grilles de catalogage. Je vous conseille de peaufiner la première grille avant de créer les autres.

Pour l'instant, on ne s'occupe pas des autorités.

Sélectionnez la grille de catalogage par défaut, et modifiez les champs les uns après les autres.

Pour chaque champ, vous pouvez préciser :

- s'il est répétable ou non. S'il est noté répétable, le signe apparaîtra devant le champ, vous permettant de le répéter lorsque vous en avez besoin.
- s'il est obligatoire ou non. Si le champ est obligatoire, vous ne pourrez valider la notice avant d'avoir saisi au moins un sous-champ dans le champ.
- Le libellé devant le champ est également défini dans les tables de paramétrage. Vous pouvez le modifier pour utiliser le libellé que vous souhaitez. Par défaut, le libellé est celui de la norme UNIMARC.

Pour chaque sous-champ, vous pouvez préciser :

- Si le sous-champ est actif ou non. Choisissez un onglet différent de -1 (ignore) pour l'activer. Notez que tous les sous-champs d'un champ donné doivent impérativement être dans le même onglet. Sinon, et si le champ est répétable, Koha ne saura comment réagir lorsque vous demanderez à répéter le champ !
- Si le sous champ est obligatoire ou non. S'il est obligatoire, vous ne pourrez valider la notice avant de l'avoir rempli.
- Si le sous champ est répétable ou non. Si un sous-champ est répétable, vous pouvez simplement le répéter en séparant les différentes valeurs répétées par le signe |
- Le "champ Koha" relié au sous-champ MARC. Koha étant multi-MARC, il faut lui "apprendre" le sens de certains champs MARC spécifiques. Par exemple, lui expliquer que le sous champ 200\$a contient le titre de la notice. Que le sous-champ 200\$f contient l'auteur... Dans la grille fournie par défaut, la plupart des "connexions" sont déjà actives. Vous pouvez éventuellement avoir à en modifier certaines comme le lien vers subject.bibliosubject, qui peut pointer vers un sous-champ 6xx, ou le lien vers additionalauthors.authors, qui contient le lien vers les co-auteurs, voire le lien vers biblisubtitle.subtitle, qui contient les sous-titres.

Les "champs Koha" peuvent provenir des tables biblio, biblioitems, items, additionalauthors, biblisubtitle, bibliosubject.

- Les "champs connexes". Lorsque vous faites une recherche sur un champ donné, Koha fera automatiquement une recherche vers les "champs connexes" que vous aurez déclaré. Cela permet d'étendre une recherche sur l'auteur aux co-auteurs, aux autorités auteurs si vous en avez... Vous pouvez étendre une recherche sur le titre aux sous-titres, au titre de collection, aux titres uniformes...
- cocher ou non l'option "caché". Cette option est normalement à utiliser pour les champs \$9 des autorités (voir chapitre sur les autorités). Si vous la cochez, le champ sera visible dans l'éditeur MARC, mais pas dans la visualisation de la notice. Vous pouvez aussi utiliser cette option pour faire disparaître un sous-champ de l'OPAC.
- cocher ou non l'option "URL". Si vous la cochez, le champ sera un lien cliquable.

Il y a également 3 options qui permettent de préciser les contraintes de saisie sur les sous-champs :

- Authorised values
- Thésaurus
- Plugin

Vous pouvez laisser ces éléments, nous y reviendrons ultérieurement.

Vous pouvez ensuite aller dans le menu catalogage, et ajouter une nouvelle notice vierge. Vérifiez là que vous avez bien activé ce que vous souhaitez.

2.2.3. Vérification de la "connexion MARC <=> non MARC"

Vous pouvez vérifier rapidement les connexions entre les bases MARC et non MARC à l'aide de

```
Koha >> Liens Koha <=> MARC-DB
```

Cet écran vous montre en un coup d'oeil les connexions entre tous les champs de la partie non-MARC et le champ/sous-champ MARC.

Vous pouvez également modifier les liens depuis cette interface, mais attention : **chaque modification est faite dans TOUTES les grilles de catalogage.**

2.3. Premier import

Retournez sur votre shell, et lancez le premier import :

```
$KOHHA/scripts/misc/bulkmarcimport.pl -d -c UNIMARC --file notices.iso
```

le -d indique de supprimer les notices existantes. Il est inutile pour le 1er import, mais comme il va être utile ensuite, autant prendre de bonnes habitudes.

le -c UNIMARC indique l'ENCODAGE des caractères dans lequel se trouve votre fichier. Attention, vous pouvez avoir des notices au format UNIMARC mais un encodage de caractères issu de la norme MARC21. Si lorsque vous regardez les notices dans Koha les accents apparaissent bizarrement, relancez l'import avec l'autre option (-c MARC21)

2.4. Premiers résultats

2.4.1. Vérification visuelle

Lorsque le script est terminé ou que vous l'avez interrompu, vous pouvez aller voir le résultat sur le catalogue.

En général, ce ne sera pas terrible. Cherchez un terme utilisé très fréquemment, et regardez si vous trouvez des notices d'exemplaires (dans la grille simplifiée comme dans la grille complète les exemplaires sont bien en évidence dans un "bloc" spécifique.

S'il n'y en n'a pas, vous devez reprendre le paramétrage de Koha pour les faire apparaître.

Vérifiez aussi que les grilles "simplifiées" (non-MARC) et "complètes" (MARC) sont bien remplies.

S'il vous semble qu'il manque des sous-champs dans la grille MARC (en visualisation), deux cas sont possibles :

- Le sous-champ est bien dans le catalogue, mais n'apparaît pas. Cela signifie que le sous-champ n'est pas "activé" mais dans l'onglet -1 (ignore). Vous pouvez aller modifier le paramétrage et retourner visualiser le résultat, sans avoir besoin de réimporter les données.
- Le sous-champ n'est effectivement pas dans la base, mais vous pensez qu'il devrait s'y trouver. Vérifiez à l'aide de l'outil dumpmarc.pl qu'il se trouve bien ou vous l'attendez.

Si vous voyez dans la notice "complète" (MARC) des données qui ne se trouvent pas dans la notice "simplifiée" (non-MARC), c'est peut-être normal, peut-être pas.

En effet, le nombre de champs de la partie "non-MARC" est limité. On peut y trouver seulement :

- author (l'auteur)
- title (le titre)

- unititle (le titre uniforme)
- notes (les notes biblio)
- abstract (le résumé)
- seriestitle (le titre de périodique/de collection)
- copyrightdate (la date de copyright)
- volume (le volume)
- number (le numéro de volume)
- classification (un code de classification local)
- itemtype (le type de document)
- url (l'URL)
- isbn et issn
- dewey
- publicationyear (la date de publication/d'édition)
- publishercode (le nom de l'éditeur)
- volumedate (la date du volume)
- volumeddesc (la description du volume)
- illus (l'illustrateur)
- pages (le nombre de pages)
- bnotes (une seconde zone de notes)
- size (la taille, c'est une zone de texte, donc 24x30 est une valeur possible)
- lccn (la classification LoC)

Toutes ces zones sont **NON** répétables dans la partie "non-MARC" de la base, partie qui apparaît dans la vue simplifiée. Si la zone est répétée, seule la première valeur sera affichée, ou, dans certains cas, les zones seront séparées par le caractère |

Il y a également 3 zones supplémentaires, qui sont répétables :

- les auteurs supplémentaires (additionalauthors.author)
- les sous-titres (bibliosubtitle.subtitle)
- l'indexation matières / sujet (bibliosubject.subject)

2.4.2. Requête de vérification.

Vous pouvez aussi lancer des requêtes SQL sur la base pour effectuer quelques vérifications.

La requête suivante affiche tous les champs et sous champs avec leur usage. C'est utile pour détecter les erreurs de paramétrage de la grille de catalogage :

```
SELECT tab, tagfield, tagsubfield, count( * ) AS tot FROM marc_subfield_structure
```

2.4.3. Itérations

Vous pouvez itérer les imports sur le catalogage jusqu'à avoir un résultat qui vous convienne.

Notez que le script `bulkmarcimport.pl` ne modifie en rien les notices bibliographiques.

Une migration est probablement une bonne raison de faire du nettoyage dans les notices, pour supprimer des choses inutiles ou en modifier d'autres pour être plus conformes à la norme.

Modifier le script `bulkmarcimport.pl` demande des connaissances assez approfondies en matière de Perl, et notamment dans la manipulation des notices MARC.

La manipulation des notices MARC passe par le package `Perl MARC::Record`.

Vous en trouverez la description à l'aide de `Perldoc` :

```
perldoc /usr/lib/perl5/site_perl/5.8.3/MARC/Record.pm
```

(le chemin est celui de ma machine Mandrake 10.0, il peut être différent selon votre distribution linux/unix)

Vous pouvez aussi trouver la même documentation sur le net :

<http://marcpm.sourceforge.net/MARC/Record.html>

et

<http://marcpm.sourceforge.net/MARC/Field.html>

`MARC::Record` permet de faire n'importe quelle manipulation sur la notice. Son seul défaut est la complexité de son interface. Mais cette complexité est imposée par la complexité de la norme iso2709.

2.5. Amélioration de la grille de saisie

Jusqu'à maintenant, les sous-champs ont tous le même format de saisie : le sous-champ est libre, le catalogeur peut saisir n'importe quelle valeur.

Dans Koha, il y a 3 autres possibilités pour la saisie d'un sous-champ :

- Liste de valeurs autorisées
- Thésaurus/autorité
- Plugin

2.5.1. Liste de valeurs autorisées

Les listes de valeurs autorisées permettent de définir les valeurs possibles pour un sous-champ donné. C'est particulièrement utile pour les sous-champs qui ne peuvent contenir selon la norme qu'un nombre

limité de valeurs.

On peut citer par exemple :

- Les langues
- Les pays
- Les codes de fonction pour les responsabilités secondaires

Nous allons voir comment activer les valeurs autorisées pour les langues.

Lorsque ce sera fait, le sous-champ ne sera plus une zone de saisie libre, mais une liste déroulante.

Note

du point de vue ergonomique, rechercher une valeur dans une liste déroulante n'est efficace que s'il y a un nombre limité de valeurs possibles. Une vingtaine est un maximum efficace. Ne multipliez donc pas inutilement les valeurs autorisées.

2.5.1.1. Définition de la liste de valeurs autorisées

Koha >> Valeurs autorisées >> Nouvelle catégorie

Saisissez alors le code de la catégorie. Vous devriez mettre un code compréhensible (LANG par exemple).

Après avoir saisi le code catégorie, vous devez également saisir la première valeur possible.

Il y a deux éléments dans les valeurs autorisées : le code qui est mis dans la notice, et le libellé qui est affiché dans la grille de catalogage.

Pour les langues, on peut saisir, par exemple *eng* comme code et *anglais* comme libellé. Le code *eng* sera entré dans la notice si vous choisissez "anglais" dans la liste.

2.5.1.2. "Connexion"

Une fois que votre liste est complète (ou dès que vous le souhaitez, vous pouvez toujours modifier la liste ensuite), retournez dans le paramétrage de la grille de catalogage, dans le champ que vous voulez "contraindre" (dans l'exemple des langues, le 101 (\$a pour la langue du document, ou autre -mais ce document n'est pas un cours UNIMARC ;-))

Dans les contraintes de saisie du champ, vous avez, dans la liste des "valeurs autorisées", la catégorie "LANG". Choisissez cette catégorie, et c'est tout, la grille de catalogage ne vous présentera plus maintenant une zone libre mais la liste des valeurs que vous aurez définies.

2.5.1.3. Deux trucs complémentaires

Il y a deux trucs complémentaires que vous devez connaître :

- Si le sous-champ est facultatif, la valeur vide est automatiquement rajoutée par l'éditeur MARC. S'il est obligatoire, elle n'est pas ajoutée, et vous ne devriez évidemment pas la saisir. Donc, dans un cas comme dans l'autre, la valeur vide est inutile !
- Vous devez savoir que les valeurs sont présentées dans l'ordre des libellés, et pas dans l'ordre des

codes. Vous devez aussi savoir que l'espace est "plus petit" que la lettre A. Vous pouvez utiliser ces deux éléments pour définir la valeur par défaut pour un sous-champ : saisissez un espace devant le libellé, et la valeur apparaîtra en premier. De plus, l'espace seul en début de chaîne de caractère étant ignoré en HTML, le libellé apparaîtra tout à fait correctement à l'écran, sans l'espace supplémentaire !

2.5.1.4. Conclusion

Maintenant que vous savez définir les listes de valeurs autorisées, vous pouvez apprendre qu'il existe un script qui fait cela pour vous pour les listes de langues.

Il s'agit du script :

```
$KOHA/misc/migration_tools/buildLANG.pl
```

Lancez le sans paramètre pour avoir les options (et n'oubliez pas le export PERL5LIB pour que le script sache retrouver certains modules spécifiques de Koha)

2.5.1.5. Cas particulier : itemtypes et branches (IMPORTANT)

La liste des valeurs autorisées lors de la "connexion" d'un sous-champ est automatiquement enrichie de deux types particuliers.

Ces deux valeurs autorisées doivent impérativement être connectées à des champs MARC

2.5.1.5.1. itemtypes

Il s'agit de la table des types de document qui sont gérés dans l'application. Cette table sert en plusieurs endroits

- Les lecteurs peuvent limiter leurs recherches à un certain type de document
- les règles de prêt dépendent des types de document.

il est donc impératif que cette table soit connectée. En UNIMARC, elle devrait logiquement être connectée au champ 200\$b.

Le champ auquel la table itemtypes est connectée doit aussi être relié au *champ Koha* `bibli-oitems.itemtype`.

2.5.1.5.2. branches

Il s'agit de la table des annexes de la bibliothèque. Il faut impérativement déclarer une annexe au moins (l'annexe principale).

Cette annexe doit être connectée dans deux sous-champs du champ des exemplaires. Pour l'UNIMARC et la recommandation 995, ce devrait être

Ces deux sous-champs doivent être reliés aux champs `items.holdingbranch` et `items.homebranch`.

2.5.2. Thésaurus / listes d'autorités

Je ne vous ferai pas l'affront de vous expliquer ce que sont les autorités selon la norme UNIMARC.

Koha sait gérer les autorités au format MARC (UNIMARC ou une autre déclinaison). Le présent

chapitre vous apprendra comment paramétrer les différentes catégories de thésaurus/autorités, et comment migrer un thésaurus.

2.5.2.1. Précision initiale

Dans la norme UNIMARC, le sous-champ \$3 de la notice bibliographique est utilisé pour stocker le numéro de la notice d'autorité, et faire ainsi le lien entre autorités (A) et bibliographies (B).

Koha sait gérer ce champ, mais utilise le sous-champ \$9 pour conserver en interne les liens entre notices (B et A). Le \$9 étant réservé au système local, c'est donc conforme à la norme. Cela présente plusieurs autres avantages de manière interne.

Notez également que la migration d'une version 2.0 vers une version 2.2 n'est pas abordée ici. Elle est assez complexe, faites moi signe si besoin, j'ai quelques scripts que je pourrais vous envoyer sans difficulté.

2.5.2.2. Définition d'un thésaurus / liste d'autorité

Lorsque vous avez installé Koha, vous avez eu la possibilité d'importer la définition des autorités UNIMARC (vers la fin de l'installation, lorsque vous avez la possibilité d'importer des fichiers SQL).

Si vous ne l'avez pas fait, faites le manuellement en recherchant le fichier SQL dans le répertoire où vous avez décompressé Koha.

Ce fichier de paramètre n'est pas exploitable directement : il fournit tous les champs de base pour que vous puissiez définir vos propres structures d'autorité.

Nous allons étudier comment mettre en place une liste d'autorité auteurs noms propres. Le principe est le même pour les autorités matière ou autres auteurs.

2.5.2.3. Paramétrage de la grille d'autorité

Allez dans Koha >> paramètres >> Structure thésaurus

Cliquez sur *Ajout d'un type d'autorité*.

Saisissez :

- Un code pour ce type d'autorité. Pour les Noms Propres, vous pouvez saisir NP
- Description : il s'agit de la description du type d'autorité. Cette description est seulement informative.
- Résumé : saisissez ici les éléments qui permettront de définir l'affichage des notices dans les listes de résultats. Vous pouvez indiquer dans cette zone les différents sous-champs qui apparaîtront, entourés par des crochet [], avec une chaîne qui peut le précéder et une chaîne qui peut le suivre. Par exemple, pour les autorités NP, vous pouvez indiquer par exemple :

```
[ 200a ] [ 200b ] [ 200c ]  
[ 400a ] [ 400z ]  
[ 100a ]
```

L'affichage du résumé de la notice dans les listes de recherche comportera ainsi la forme retenue et la ou les formes rejetées. Si vous trouvez que cela fait des listes trop longues, ne mettez que la ligne avec les [200].

- Le numéro (000 - 999) du champ de la notice d'autorité qui est reporté dans la notice bibliograph-

ique. Pour les autorités NP, par exemple, c'est le champ 200 qui sera reporté dans la notice bibliographique (dans le champ 700,701 ou 702) Comme vous le constatez, vous n'indiquez qu'un champ, pas des sous-champs. En effet, Koha reportera automatiquement dans la notice bibliographique tous les sous-champs. Dans notre exemple, le 200\$a ira dans le 700\$a, le 200\$b ira dans le 700\$b, ...

Lorsque vous avez terminé, validez la saisie. Le type d'autorité devrait apparaître dans la liste des autorités. Cliquez ensuite sur "structure MARC". La première fois, Koha détectera qu'il n'y a pas encore de grille pour ce type d'autorité et vous demandera quelle grille existante il doit recopier. Au début vous ne pourrez choisir que le type "default" bien évidemment.

Vous pouvez alors mettre au point la grille de catalogage de l'autorité de la même manière que vous avez défini la grille de catalogage des bibliographies. Certaines options spécifiques aux notices bibliographiques ne sont pas disponibles, mais vous devriez retrouver des choses connues.

Lorsque la grille d'autorité est bien définie, vous pouvez aller dans le menu autorités et créer une notice d'autorité. vous verrez ainsi à quoi ressemble votre grille de saisie d'autorité.

2.5.2.4. Connexion B => A

Maintenant que les notices d'autorités sont paramétrées, il faut retourner dans le paramétrage des notices bibliographiques pour relier les notices B (bibliographies) et A (autorités).

Pour cela, il faut retourner dans

```
Koha >> Paramètres >> biblio structure
```

Allez sur le champ 700, qui contient la forme d'autorité de l'auteur principal (ou des auteurs principaux).

2.5.2.4.1. sous-champ \$9

Ainsi que nous l'avons déjà expliqué, Koha utilise le champ \$9 pour faire le lien entre la notice B et la notice A.

Vérifiez donc que le sous-champ \$9 est créé, et s'il ne l'est pas, créez le, et activez le dans le même onglet que les autres sous-champs du champ 700. Vous devriez cocher la case "caché" pour qu'il n'apparaisse pas dans l'OPAC puisqu'il est inutile.

2.5.2.4.2. Connexion

Allez modifier les sous-champs du champ 700. Sur l'un d'entre eux (logiquement, le \$a, mais ce n'est pas obligatoire), sélectionnez comme contrainte de saisie, dans la liste des autorités/thésaurus, la valeur NP.

Vous venez de dire à Koha que le champ 700 contient des valeurs qui sont issues de la liste d'autorité NP.

Vous pouvez aller ajouter une notice. Devant le 700\$a, vous verrez qu'il y a maintenant trois points ...

Ces ... sont un lien cliquable qui ouvre une fenêtre "popup", qui vous permet de faire une recherche d'autorité et reporter ce que vous aurez trouvé. Lorsque vous ferez un report, la "popup" se fermera, et les valeurs de la notice d'autorité que vous avez sélectionnée seront reportées automatiquement dans la notice bibliographique.

2.5.2.5. Reconstruction automatique

Il n'est pas possible de proposer un script de reconstruction qui fonctionne à tous les coups. Surtout qu'il devra se comporter différemment pour les NP (Noms Propres), les NC (Noms Communs), les CO

(Collectivités)...

Vous trouverez néanmoins dans le répertoire misc/migration_tools un script pour reconstruire les autorités NC :

```
build6xx.pl, qui reconstruit le 606, avec la gestion des $x.
```

Vous pouvez l'utiliser comme base pour reconstruire les 700, les 500 ce dont vous avez besoin.

2.5.3. Plugins

Les plugins sont des modules complémentaires qui permettent de réaliser toutes sortes de traitements pour aider à la saisie des notices bibliographiques (ou d'autorité)

Par exemple, pour UNIMARC, vous trouverez des plugins qui vont vous simplifier la vie pour :

- la saisie des éditeurs et des collections
- la saisie des champs codés (1xx)

2.5.3.1. Principe général

Un plugin s'active sur un sous-champ donné. Lorsqu'il est actif, le sous-champ a les propriétés suivantes :

- Il peut être rempli manuellement comme un champ normal
- Il dispose de ... qui ouvrent une popup. Cette popup vous assistera dans le remplissage du champ
- Lorsque vous arrivez ou quittez le sous-champ, certains plugins vont automatiquement aller modifier des valeurs dans d'autres sous-champs de la grille de catalogage.

2.5.3.2. Champs codés

Dans la version 2.2 de Koha, l'Ecole Nationale Supérieure des Mines de Paris a développé un plugin pour tous les champs 1xx, qui sont les champs codés.

Les ... ouvrent une popup qui présente pour chaque élément codé du sous-champ des listes déroulantes, permettant de le saisir de manière complète assez aisément.

2.5.3.3. Editeurs/collections

Les plugins appelés unimarc_field_210c.pl et unimarc_field_225a.pl vont remplir automatiquement les champs d'éditeur et de collection à partir de l'ISBN (s'il existe).

Ce plugin est assez délicat à configurer.

Tout d'abord, il utilise... les fichiers d'autorité. La liste des éditeurs peut en effet être vue comme une liste d'autorité (=la forme sous laquelle est noté l'éditeur)

Donc : Koha >> paramètres >> Structure des autorités

```
Ajoutez un type d'autorité appelé impérativement EDITORS.  
Le résumé sera [200a ][ / 200b]
```

le champ de report sera 200

Définissez pour ces autorités éditeurs la structure suivante :

```
200$a => ISBN
200$b => Editeur
200$c (répétable) => collections
```

Allez ensuite "connecter" les sous champs UNIMARC 210\$c et 225\$a.

C'est terminé.

Vous pouvez saisir les "notices d'éditeur" en prenant pour ISBN les 2 premières parties de celui-ci (1-22-333333-4, prenez 122), mais SANS les tirets.

Mettez ce que vous voulez dans l'éditeur, et les collections (en les séparant par | comme d'habitude pour répéter un sous-champ.

Ces fonctionnalités sont très utiles pour les bibliothèques qui ont un nombre limité d'éditeurs (les bibliothèques spécialisées)

Lorsque vous allez cataloguer une notice, après la saisie de l'ISBN, allez sur l'éditeur, il sera rempli automatiquement si vous avez tout rentré correctement.

Cliquez ensuite sur les ... en face des collections, et vous pourrez choisir parmi les différentes collections qui existent pour cet éditeur.

Notez qu'il n'est pas encore possible de créer une collection à la volée, ni de fabriquer automatiquement les tables d'autorités à partir des notices (lors d'une migration par exemple), mais quelqu'un le fera bien un jour.

2.6. Cas de la migration d'un fichier non iso2709

Ce cas est très simple à gérer : il faut construire une notice MARC::Record (donc iso2709) à la volée.

C'est assez complexe. Et surtout tous les cas sont envisageables !

2.6.1. Migration d'un fichier Texto

Je dispose de quelques routines pour migrer une base Texto. Je ne les mets pas ici parce que chaque installation de Texto A ses propres déclarations de champ.

Mais l'exploitation d'un fichier d'ajout piloté Texto peut se résumer ainsi :

```
$/ = "";
while (<AJOUT_PILOTE>)
{
    my @fichier = split/\n/, $_;
    # print "taille fichier". $#fichier. "\n";
    foreach my $ligne (@fichier) {
        # on examine le contenu de la fichier
        if ($ligne eq "") {# separateur de fichiers
        }
    }
    my @mots = split(/\./, $ligne);
    $mots[0] =~ s/ //g;
    $last=$mots[0] if $mots[0];
    $resul{$last} .= $mots[1]. " ";
}
```

```
}  
  
# ok, on a l'enregistrement, on construit le MARC::Record.  
my $newRecord = MARC::Record->new();
```

Ce bout de code va construire une table de hashage pour chaque notice. On peut ensuite en faire une notice iso2709, et l'importer normalement

```
$resul{ISBN} =~ s/-//g;  
my $newField = MARC::Field->new(  
    '010', '', '',  
    'a' => $resul{ISBN},  
);  
$newRecord->insert_fields_ordered($newField);  
$newField = MARC::Field->new(  
    '011', '', '',  
    'a' => $resul{ISSN},  
);  
newField = MARC::Field->new(  
    '200', '', '',  
    'a' => $resul{TIT},  
    'b' => $resul{TYP},  
    'e' => $resul{STI},  
    'f' => $resul{AUT},  
    'g' => $resul{NOT},  
);  
$newRecord->insert_fields_ordered($newField);  
my ($bibid,$oldbibnum,$oldbibitemnum) = NEWnewbiblio($dbh,$newRecord,'') unless ($
```

Notez que le code qui précède ne traite pas des notices d'exemplaire (là, c'est vraiment trop spécifique)