

**Ce document N'est PAS libre de droit.**

Pour tout renseignement contacter :

Yves AUBIN

Directeur

Conservateur en chef

La Bibliothèque

Ville de Saint-Herblain

02 28 25 25 53

[yves.aubin@saint-herblain.fr](mailto:yves.aubin@saint-herblain.fr)

<http://la-bibliotheque.saint-herblain.fr>

OU

Quentin CHEVILLON

Responsable du Département de l'Inform  
atique Documentaire :

Responsable du Domaine "Généralités"

Bibliothécaire :

02.28.25.25.41 :

[quentin.chevillon@saint-herblain.fr](mailto:quentin.chevillon@saint-herblain.fr) )

# OPAC 2.0

## Manuel de maintenance

2 mai 2007

## I. Préambule

Ce document a pour objectif de synthétiser le fonctionnement et les grandes lignes de développement de l'OPAC 2.0 afin que ce dernier puisse continuer à être maintenu si je venais à quitter Saint-Herblain.

La lecture de ce document suppose :

- Une bonne connaissance de PHP
- Une bonne connaissance de Flash et ActionScript
- Des notions sur les WebServices ([http://fr.wikipedia.org/wiki/Service\\_Web](http://fr.wikipedia.org/wiki/Service_Web)) et sur les normes et protocoles qui y sont liés, à savoir SOAP (<http://fr.wikipedia.org/wiki/SOAP>), XML ([http://fr.wikipedia.org/wiki/Extensible\\_markup\\_language](http://fr.wikipedia.org/wiki/Extensible_markup_language)) et HTTP (<http://fr.wikipedia.org/wiki/HTTP>)

## II. Architecture générale

L'information, dans le nouvel OPAC, suit un parcours assez complexe constitué de nombreuses étapes : au total, 5 machines sont concernées :

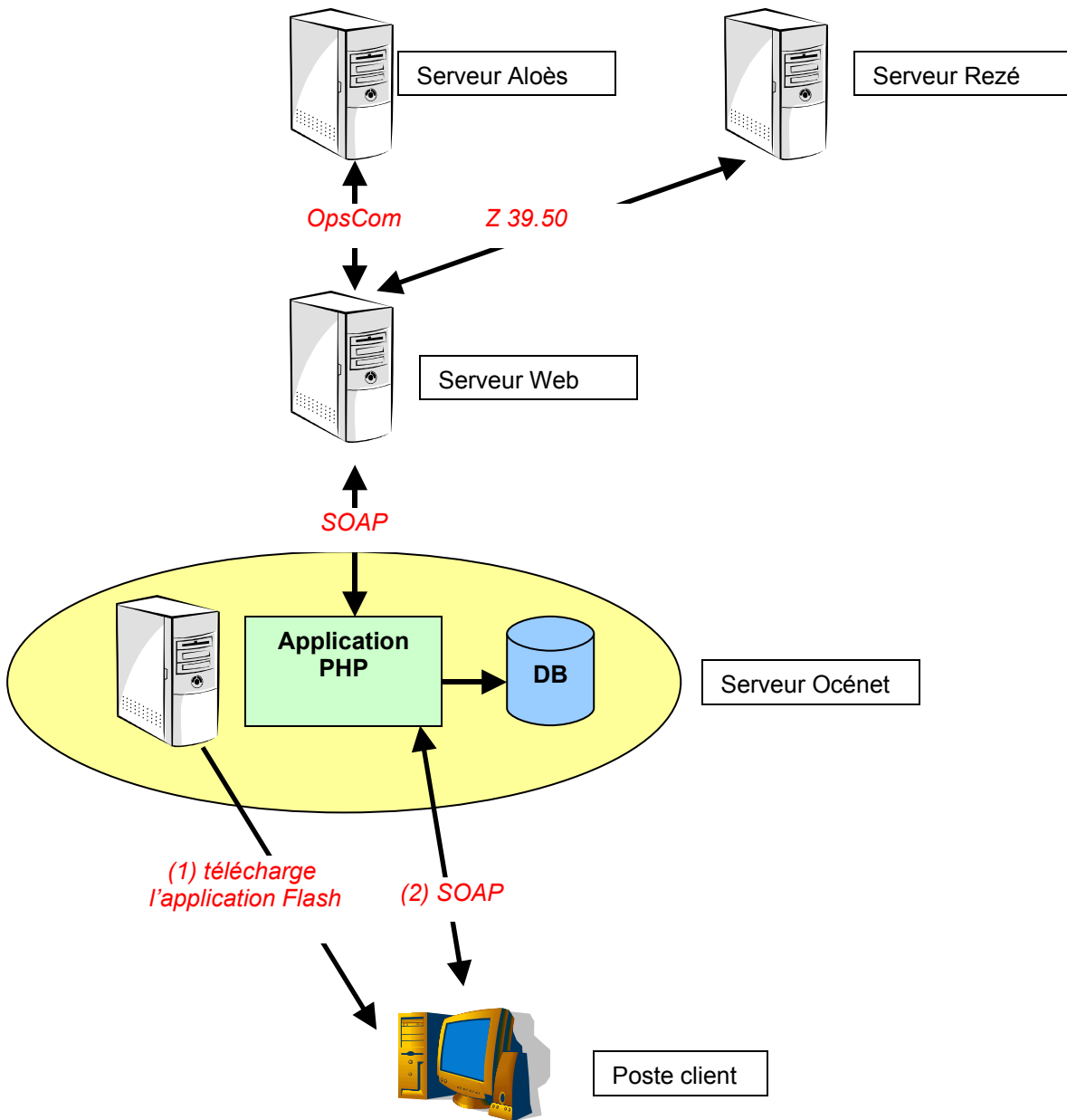
- Le **PC client** qu'utilise l'internaute qui consulte l'OPAC (ce peut être également un des postes dédiés au public de la bibliothèque).
- Le **Serveur Océanet** qui héberge le site mais également les applications développées et certaines bases de données liées à la partie '2.0' (base des commentaires, suggestions...)
- Le **Serveur Web** hébergé en mairie.
- Le **Serveur Aloès** hébergé à la bibliothèque.
- Le **Serveur Web et Z 39.50 de Rezé**.

L'information circule entre ces différentes machines selon des normes et des formats différents. Voici un synoptique du fonctionnement.

1. Tout d'abord l'internaute se connecte au **serveur Océanet** et télécharge le site en Flash. Contrairement à des pages HTML et PHP, le site Flash est une véritable **application**. Il ne se contente pas d'afficher des données, il va lui-même aller les chercher et les synthétiser.
2. Lorsque l'internaute effectue une recherche, l'application Flash envoie la requête à une application en PHP située sur le **serveur Océanet**. Celle-ci renvoie les informations demandées en utilisant le protocole SOAP, c'est-à-dire du XML sur HTTP.
3. L'application PHP située sur le **serveur Océanet** va elle-même récupérer les informations auprès de 2 sources :
  - a. Elle va se connecter au **serveur web** pour récupérer les informations liées au catalogue (liste de notices, notice détaillée, mes prêts...). Cet échange se fait également en utilisant SOAP (HTTP + XML).
  - b. Elle va également récupérer des informations concernant les commentaires et les suggestions (les lecteurs ayant emprunté ce doc ont également...) dans une base de données locale.
4. Enfin le **serveur web** (application Opsys en ASP) va récupérer les données :
  - a. Soit auprès du **serveur Aloès** en utilisant le protocole propriétaire Opscom.
  - b. Soit auprès du **serveur Z 39.50 de Rezé** en utilisant la norme Z 39.50.

Un des points faibles de l'application est qu'il suffit qu'un des 3 serveurs (serveur Océanet, serveur web, serveur aloès) soit défaillant ou ralenti pour que l'application soit défaillante ou ralentie.

Voici un schéma synthétisant les flux d'information :



Dans ce schéma, on voit que, à certains égards, le rôle du serveur Océanet est un peu redondant. En effet, l'application Flash interroge le serveur Océanet en utilisant SOAP, et ce dernier interroge le serveur web en utilisant ce même protocole.

En fait, l'application Flash aurait très bien pu interroger elle-même le serveur web, mais pour des raisons pratiques, j'ai préféré passer par l'intermédiaire d'une application en PHP. En effet, les WebServices fournis par Opsys sont assez complexes à utiliser et d'autre part, Il n'est pas très facile de gérer les WebServices et XML en Flash. Aussi, j'ai préféré créer une couche intermédiaire en PHP qui transforme les WebServices 'compliqués' d'Opsys en WebServices plus 'simples' d'utilisation.

### III. L'Application en Flash

#### A. Présentation générale

L'Application en Flash se compose de 3 fichiers :

- recherche\_aloes3 fla : c'est la page flash de recherche qui sera chargée dans le reste du site en flash
- recherche\_aloes\_core.as : ce script contient des fonctions actionscript qui sont utilisées pour la communication avec le serveur Océanet. C'est ici que sont gérés les webServices, l'analyse du XML et l'affichage.
- Recherche\_aloes.as : ce script contient d'autres fonctions plus généralistes.

Recherche\_aloes3 fla contient assez peu de code, il sert surtout d'architecture générale au catalogue. Les images 2 à 14 sont dédiées aux formulaires de recherche (1 image par formulaire). L'image 15 correspond à l'affichage d'une liste de notices ou d'autorités, l'image 18 correspond à l'affichage d'une notice détaillée et enfin l'image 20 correspond à l'affichage de la liste des prêts.

En ce qui concerne les calques :

- Le calque 'titre' contient le titre de chaque image (catalogue danse, catalogue musique...)
- Le calque 'navigation' contient les boutons permettant de passer à la notice précédente/suivante, au tri au nombre de résultats...
- Le calque 'image' contient à l'image 18 une zone de texte pour affichage de la couverture du document dans la notice détaillée.
- Le calque 'défilement' contient les boutons nécessaires au défilement des différentes zones.
- Les masques suggestions et masque\_suggestions permettent l'affichage des suggestions (ceux qui ont emprunté ce doc ont aussi emprunté...). Le calque suggestion contient 8 emplacements de suggestions qui peuvent défiler horizontalement, et le masque s'assure qu'on ne puisse pas en voir plus de 3 à la fois.
- Le calque 'contenu' contient à l'image 15 les zones de textes qui contiendront la liste des notices (5 documents à la fois), à l'image 18 les zones qui contiendront la notice détaillée et les commentaires et à l'image 20 les zones qui contiendront la liste des prêts.
- Le calque 'formulaires' contient les différents formulaires de recherche
- Le calque 'zones' contient les pavés de couleur qui sont sous-jacents aux différentes zones de texte dans l'affichage des résultats.
- Le calque 'menu' contient le menu général à droite de l'écran.
- Enfin il y a 4 calques contenant uniquement de l'actionscript :
  - Action\_validation\_formulaire : une fonction (une par image) est associée à chacun des formulaires. Cette fonction va analyser les données saisies dans le formulaire, générer une phrase de recherche Opsys (du type « RC = pennac ET TIT = cabot ») et appeler la fonction rechercher() qui va se charger d'envoyer cette phrase et de récupérer les données.  
Par ailleurs ce calque se charge de mettre le focus sur le premier champ de chaque formulaire et de gérer la validation du formulaire en cliquant sur 'entrée' (cf. image 1)
  - Action\_navigation : ce calque va gérer l'interaction avec la barre de navigation (notice précédente/suivante, le tri). En fait, il ne fait que détecter les clics de l'utilisateur et appeler des fonctions qui sont définies dans recherche\_aloes.as. Il gère également toute la partie défilement des différentes zones lorsque l'on clique sur les boutons de défilement.
  - Action\_menu : ce calque gère l'interaction avec le menu de droite. D'une part l'effet de 'rollover' et d'autre part le renvoi vers tel ou tel formulaire de recherche quand on clique sur un des boutons.
  - Action : contient la définition de différentes variables globales mais surtout :  
-> L'inclusion dans le script de recherche\_aloes.as et recherche\_aloes\_core.as

-> La définition de l'url du serveur à interroger ([http://www.la-bibliotheque.com/aloes\\_php.php](http://www.la-bibliotheque.com/aloes_php.php))

Pour finir, la gestion des commentaires sur une notice se fait dans le clip clip\_commentaires.

## **B. Les fonctions principales**

Les principales fonctions se trouvent dans recherche\_aloes\_core.as ce sont les suivantes :

- **Afficher\_notice()** : cette fonction permet d'afficher une notice détaillée. Elle reçoit en paramètre (entre autres) le numéro de la notice et son rang dans une liste. Elle se connecte au serveur Océanet et lui transmet ces informations et récupère en retour des données XML contenant les infos sur la notice. Elle analyse l'XML en extrait les informations et les affiche.
- **Rechercher()** : chaque formulaire de recherche appelle cette fonction qui se connecte au serveur Océanet et lui transmet la phrase de recherche. Le serveur lui retourne un certain nombre d'infos (nombre de résultats...) ainsi que les 5 1eres notices en XML. Ces données sont transmises à la fonction afficher\_liste() qui gère l'affichage.
- **Rechercher\_serie\_notices()** : cette fonction permet de rechercher une série de notices liées à une pagination. Lorsque l'utilisateur valide un formulaire de recherche, c'est la fonction rechercher() qui est appelée et qui récupère les 5 premières notices. Mais ensuite, lorsque l'on clique sur page suivante... c'est la fonction rechercher\_serie\_notices() qui est appelée, se connecte au serveur et récupère les notices suivantes. Les données XML sont transmises à la fonction afficher\_liste() qui gère l'affichage. C'est également cette fonction qui est appelée lorsque l'on trie une liste de notices.
- **Afficher\_liste()** : cette fonction est appelée par rechercher\_serie\_notices(), rechercher() ou rebondir() et permet d'afficher une liste de notices. Elle n'effectue pas de recherche (ne se connecte pas au serveur) mais reçoit en paramètre des données XML récupérées par les fonctions susdites. Elle les analyse et les affiche.
- **Rebondir()** : cette fonction récupère les notices liées à une ou plusieurs autorités donnée (un sujet, un auteur...). Elle est utilisée dans la fonctionnalité 'rebondir' mais également dans la recherche par liste quand on a coché une ou plusieurs autorités.
- **Lister\_prets()** : cette fonction permet d'afficher les paramètres d'un lecteur et la liste de ses prêts. Elle se connecte au serveur envoie les paramètres (N° de carte et mot de passe) et récupère les données XML. Elle analyse ces données et affiche le résultat.
- **Recupere\_suggestions()** : cette fonction permet d'afficher des suggestions liées à un document (les lecteurs qui ont lu ont aussi lu...). Elle se connecte au serveur envoie les paramètres (N° de notice) et récupère les données XML. Elle analyse ces données et affiche le résultat.
- **Recupere\_commentaires()** : cette fonction permet d'afficher les commentaires associés à une notice et également d'enregistrer un nouveau commentaire. Elle se connecte au serveur envoie les paramètres (N° de notice, éventuellement nouveau commentaire) et récupère les données XML. Elle analyse ces données et affiche le résultat.

## **C. La navigation**

La navigation correspond à la gestion des 3 boutons : notice précédente/suivante/retour à la liste.

Le type d'action à effectuer quand on clique sur un des boutons varie en fonction du contexte (affichage d'une liste, affichage d'une notice...).

Ce sont les fonctions principales (cf ci-dessus) qui définissent comment l'application doit réagir à un clic sur ces boutons. Prenons un exemple avec la fonction afficher\_notice() mais qui est valable avec les autres.

Admettons qu'un utilisateur ait cliqué sur une des notices dans une liste pour en afficher le détail. Il faudra alors qu'un clic sur le bouton gauche affiche la notice précédente dans la liste, un clic sur le bouton droit affichera la notice suivante, tandis que le bouton du milieu permettra de retourner à la liste.

Pour gérer cela, la fonction `afficher_notice()` va renseigner 3 tableaux qui sont des variables globales :

- `aloes_action_btright`
- `aloes_action_left`
- `aloes_action_up`

chacun de ces tableaux comprend une clef « action » et une clef « parametres ».

La fonction `afficher_notice` va donner à `aloes_action_btright.action` et `aloes_action_btleft.action` la valeur « `afficher_notice` » pour indiquer qu'il faudra afficher une notice détaillée, tandis qu'elle donnera à `aloes_action_btup.action` la valeur « `rechercher_serie_notices` » pour indiquer qu'il faudra afficher une liste de notices.

Les variables `aloes_action_btright.parametres`, `aloes_action_btleft.parametres` et `aloes_action_btup.parametres` sont chacune un tableau qui contiendra les différents paramètres nécessaires à l'action. Dans tous les cas, le `GUIDSession` (c'est un identifiant unique de session fourni par Aloès à chaque connexion). Pour afficher une notice détaillée on stockera le numéro de la notice, son rang dans la liste... Pour afficher une liste, on indiquera le numéro de la première notice et le nombre de notices à afficher...

Lorsque l'utilisateur clique sur un des boutons, il appelle les fonctions `click_bouton_left()`, `click_bouton_right()` ou `click_bouton_up()` qui se trouvent dans `recherche_aloes.as`. Ces fonctions vont analyser le contenu de `aloes_action_btright`, `aloes_action_left` et `aloes_action_up` et appeler les fonctions qui s'imposent en leur fournissant les paramètres nécessaires.

## ***D. SOAP et les WebServices***

Pour se connecter (via HTTP) au serveur Océanet, récupérer et analyser les données XML, l'application utilise la classe `actionsript XML`. Sa méthode `load()` permet d'envoyer une requête au serveur avec un certain nombre de paramètres et de récupérer les données. Elle fournit ensuite un certain nombre de méthodes permettant d'explorer l'arborescence du document XML.

## ***E. Tests et debugging***

Si l'on souhaite effectuer des tests en local avant de publier une nouvelle version, il faut modifier l'url du serveur. Cela se fait dans le calque 'actions' image 2.

## IV. L'Application en PHP

L'application en PHP est localisée sur le serveur Océanet, mais il serait également possible de l'installer sur le serveur web (cela nécessiterait d'y installer MySQL, par ailleurs de petites modifications pourraient être nécessaire du fait que PHP 4 est installé sur le serveur Océanet tandis que PHP 5 est installé sur le serveur web).

Elle se compose des fichiers suivants :

- Aloes\_php.php qui est le script principal
- Aloes2flash.php qui contient les méthodes permettant de communiquer avec le serveur web aloès en utilisant des Webservices.
- Aloes\_2\_0.php qui contient des méthodes permettant la gestion des fonctionnalités supplémentaires (affichage de l'image, suggestions, commentaires)
- Aloes\_sql.php qui contient des fonctions génériques pour faciliter les requêtes SQL
- Aloes\_div.php qui contient des fonctions utiles et génériques.
- Aloes\_tableau\_serveurs.php qui contient les paramètres des différents serveurs Z 39.50 à utiliser.
- Répertoires nusoap et domit contenant des classes externes. Nusoap permet de gérer les connexions HTTP tandis que domit permet d'analyser les données XML.

A part aloes\_php.php qui est à la racine du serveur, tous les autres fichiers se trouvent dans le répertoire aloes2flash\_include.

### A. Interaction avec l'application Flash

L'Application PHP interagit avec l'application Flash via HTTP. L'application Flash envoie des requêtes via l'URL. L'application PHP répond en retournant des données en XML dans le corps du message.

Voici un exemple d'échange :

1) L'application Flash envoie une requête HTTP :

[http://www.la-bibliotheque.com/aloes\\_php.php?operation=rechercher&phrase=RE%3Dduby](http://www.la-bibliotheque.com/aloes_php.php?operation=rechercher&phrase=RE%3Dduby)

2) L'application PHP renvoie du XML :

```
<erreur></erreur>
<reponse><liste_notices><notice><NumNotice>0242727</NumNotice>
<RegleNotice>UNI:1</RegleNotice>
<Contenu>Amour et sexualité en Occident / introd. de Georges Duby,... -
Seuil, 1991</Contenu>
<RangNotice>1</RangNotice>
<ISBN>2020125005</ISBN>
</notice>
<notice><NumNotice>0165856</NumNotice>
<RegleNotice>UNI:1</RegleNotice>
<Contenu>An 1000 An 2000 : sur les traces de nos peurs / Georges Duby. -
Textuel, 1995</Contenu>
<RangNotice>2</RangNotice>
<ISBN>2909317102</ISBN>
</notice>
<notice><NumNotice>0192222</NumNotice>
<RegleNotice>UNI:1</RegleNotice>
<Contenu>L' art et l'image : une anthologie / Georges Duby. - Parenthèses,
2001</Contenu>
<RangNotice>3</RangNotice>
<ISBN>2863646230</ISBN>
</notice>
<notice><NumNotice>0258089</NumNotice>
```

```
<RegleNotice>UNI:1</RegleNotice>
<Contenu>Atlas historique : l'histoire du monde en 334 cartes / sous la
dir. de Georges Duby,... - Larousse, 1994</Contenu>
<RangNotice>4</RangNotice>
<ISBN>2035212146</ISBN>
</notice>
<notice><NumNotice>0249868</NumNotice>
<RegleNotice>UNI:1</RegleNotice>
<Contenu>La chevalerie / raconté par Georges Duby,... - Perrin,
1998</Contenu>
<RangNotice>5</RangNotice>
<ISBN>2262014345</ISBN>
</notice>
</liste_notices>
<nb_resultats>41</nb_resultats>
<GUIDSession>d57ade1a452345ef8ec2629bdb394a161local</GUIDSession>
<IDEtape>T99999</IDEtape>
<page>0</page>
</reponse>
```

Je ne vais pas donner la syntaxe complète des requêtes et le format de l'XML retourné, reportez-vous aux fonctions les gérant. Rappelez-vous que pour faire des tests, il est possible de saisir directement les requêtes dans un navigateur et le XML s'affiche alors à l'écran.

Une technique facile à utiliser pour faire du debugging est de faire un 'trace (url)' dans l'application en flash pour afficher l'url de requête envoyée à l'application PHP. Vous n'avez plus qu'à copier cette url dans un navigateur pour voir ce que l'application retourne.

## **B. Les principales opérations**

L'essentiel du script aloes\_php.php consiste à dispatcher les différents types de requêtes en utilisant des fonctions spécialisées. Pour cela, il utilise le paramètre 'operation' fourni dans l'url :

[http://www.la-bibliotheque.com/aloes\\_php.php?operation=rechercher&phrase=RE%3Dduby](http://www.la-bibliotheque.com/aloes_php.php?operation=rechercher&phrase=RE%3Dduby)

Les différentes opérations gérées sont :

- Rechercher : recherche dans le catalogue et retourne les 5 1eres notices trouvées
- liste\_notices : récupération des notices suivantes ou précédentes après une recherche (gestion de la pagination)
- notice\_detail : notice détaillée
- rebondir : recherche des notices filles à partir des n° des notices mères
- lister\_prets : infos lecteur et listes prêts
- suggestions : suggestions liées à une notice
- commentaires : commentaires liés à une notice

### **C. Interaction avec les WebServices Aloès**

L'application aloes\_php.php va se connecter au serveur web en utilisant les WebServices Aloès pour récupérer les données demandées par l'application Flash.

Pour ce faire, elle instancie un objet aloes2flash dont la classe est définie dans le fichier aloes2flash.php. Cette classe définit des méthodes qui vont permettre de :

- traiter la requête
- se connecter au serveur web via les WebServices Aloès
- lui envoyer une ou plusieurs requêtes conformes à la syntaxe définie par Opsys
- récupérer les données en XML
- les analyser et les reformater
- retourner les informations demandées par l'application flash en XML également.

En général, pour traiter une requête venant de l'application flash, il faut souvent envoyer plusieurs requêtes au serveur web.

Les méthodes de la classe aloes2flash peuvent être classées en 2 grandes catégories :

- les méthodes 'requete\_xxx' sont de bas niveau : elles génèrent la requête XML vers le serveur web et transmettent le résultat en gérant au passage l'intégrité des paramètres et les erreurs.
- les méthodes 'lance\_xxx' sont de haut niveau : elles ne gèrent pas la connexion HTTP avec le serveur web mais se contentent de préparer les paramètres de la requête, de récupérer et traiter les infos en XML puis de générer la réponse en XML.

La connexion HTTP elle-même est gérée par la méthode 'requete\_soap()' qui utilise le package nusoap.

### **D. Les fonctionnalités 2.0**

Le fichier aloes\_2\_0.php contient des fonctions qui gèrent les fonctionnalités 2.0 à savoir les suggestions et les commentaires. Pour ce faire elles utilisent des bases de données locales contenant des informations exportées périodiquement d'Aloès (pour les suggestions) ou saisies par les usagers (pour les commentaires).

Pour les suggestions on a la table 'prets' (qui contient un historique anonyme de prêts associant un code lecteur à un numéro de notice) et la table 'notices' qui contient le détail de chaque notice ainsi que des critères de restriction.

Pour les commentaires, on a la table 'commentaires'.

### **E. Tests et debugging**

Pour faire des tests, il peut être nécessaire d'utiliser ces scripts en local avec une base de données installée sur ça machine. Pour ce faire, il suffit de modifier les paramètres au début de aloes\_php.php.

## V. Le Serveur Web et les WebServices Aloès

Les WebServices Aloès sont développés en ASP.net et se trouvent sur le serveur Web hébergé en mairie. Il y en a en fait 2 : l'un gère les requêtes 'classiques' l'autre les requêtes de type Z 39.50.

Vous pouvez avoir une documentation succincte de ces WebServices aux adresse :

<http://83.145.103.118/WebSrvAloes/ServiceRecherche.asmx>  
<http://83.145.103.118/WebSrvz3950/webz3950.asmx>

et une description détaillée du XML utilisé ici :

<http://83.145.103.118/WebSrvAloes/ServiceRecherche.asmx?WSDL>  
<http://83.145.103.118/WebSrvz3950/webz3950.asmx?WSDL>

Il n'y a en revanche aucune application 'maison' installée sur ce serveur. Il serait tout à fait possible, en revanche, si nous le désirions d'installer l'application PHP sur le serveur web plutôt que sur le serveur Aloès.

**NOTE** : En cas de changement ou de réinstallation du serveur, il se peut que l'affichage des notices soit altéré et que les fonctionnalités avancées ne fonctionnent plus. C'est parce qu'il faut autoriser le serveur web à utiliser le format ISBDFLASH défini sur le serveur Aloès.

La procédure est si complexe qu'il vaut mieux demander à la maintenance Opsys d'effectuer cette opération.

## VI. Paramétrages sur le Serveur Aloès

L'affichage de la notice et des exemplaires est essentiellement paramétrable via le serveur Aloès (module d'administration). Tout ce qui suit peut être effectué par la maintenance Aloès, mais il faut leur fournir les bonnes informations.

### A. La notice

L'affichage de la notice se paramètre dans le module d'administration >> Catalogue >> Paramètres >> Saisie des ISBD.

Dans l'onglet 'général' choisir le format Unimarc et le nom de l'affichage 'ISBD Flash – affichage OPAC Flash'. Aller ensuite dans l'onglet 'Détail'.

Il est possible ici de modifier l'affichage de la notice en visualisant chaque bloc (déroulant en haut à gauche).

**ATTENTION** : 2 blocs sont importants car ils servent non pas à l'affichage, mais à fournir des informations spécifiques à l'application Flash.

Le 1<sup>er</sup> bloc DOIT contenir le champ 010\$a et DOIT contenir en séparateur fin « AMAZONASIN ». Cela permet à l'application Flash d'afficher les couvertures des livres à partir de l'ISBN.

De même, un des blocs (actuellement le n° 26 mais ça peut changer) est utilisé pour gérer le champ 990\$a (uniquement pour les notices Z 39.50). Ce bloc doit contenir « FLASHEXEMPLAIRE » en séparateur début pour que l'application PHP puisse extraire les données d'exemplaires du corps de la notice.

### B. Les exemplaires

L'affichage des exemplaires (sauf Z 39.50) est géré par l'affichage 'ISBDOPACEXE' (auquel on accède en suivant la même méthode que ci-dessus).

### C. Disponibilité

L'affichage de la disponibilité (disponible, non disponible retour prévu le ...) est géré par un ligdet.

On y accède en allant dans le module d'administration >> Outils >> LIGDET

Choisir le ligdet AFFEXEDETAIL1 et aller dans l'onglet détail pour le modifier.