



Sommaire

A)Solution technique utilisée.....	3
B)Organisation des fichiers et répertoires.....	4
1)Scripts d'administration.....	4
2)Dossiers « html ».....	4
3)Dossiers « includes ».....	6
4)Classes utilisées dans VCOOP.....	7
Classe de base.....	7
Articles (items).....	7
Sources.....	8
Alertes.....	8
Newsletters et éditions.....	8
Utilisateurs et droits d'accès.....	8
Autres fonctions et classes.....	9
5)Gabarits.....	9
C)Procédure d'installation.....	11
1)Mise en place des scripts PHP.....	11
2)Contenu de la base de données.....	11
3)Configuration de VCOOP.....	11
4)Configuration de PhpBB.....	12
5)Configuration d'Apache.....	12
6)Configuration de MngoSearch.....	13
7)Scripts périodiques.....	14
D)Description des tables spécifiques à VCOOP.....	15
1)TABLE source.....	15
2)TABLE item.....	16
3)TABLE selection.....	17
4)TABLE commentaire.....	17
5)TABLE scommentaire.....	17
6)TABLE item_rubrique.....	17
7)TABLE source_rubrique.....	18
8)TABLE profil.....	18
9)TABLE paragraphe.....	19
10)TABLE typedocs.....	19
11)TABLE newsletter.....	19
12)TABLE npartie.....	20
13)TABLE edition.....	20
14)TABLE segment.....	21
15)TABLE newsletter_pending_subscribe.....	21
16)TABLE newsletter_subscribe.....	21
17)TABLE alerte_item.....	22
18)TABLE alerte.....	22
19)TABLE forum_group_owner.....	22

A) Solution technique utilisée

VCOOP repose sur un certain nombre de composants issus du monde du logiciel libre. Les principaux composants utilisés étant :

- Le serveur Web : Apache.
- Le langage : PHP (avec notamment l'extension MnegoSearch activée).
- L'interface d'abstraction d'accès aux bases : ADODB (le développement étant principalement effectué sur Mysql).
- Le serveur de bases de données : Mysql (certaines requêtes sont relativement spécifiques, mais l'adaptation à un autre moteur de bases ne devrait pas être très compliqué).
- Le langage pour les gabarits : Smarty.
- La gestion des droits d'accès : LiveUser.
- L'interprétation des flux RSS : Magpie.
- L'éditeur Wysiwyg HTML : HtmlArea.
- L'affichage des arbres en JavaScript : jscooktree.
- Le moteur de recherche : MnegoSearch.
- Le gestionnaire de forums de discussion : phpBB.
- Le gestionnaire de listes de diffusion : Sympa.

D'autres classes plus secondaires sont également utilisées, notamment comme dépendances de celle citées précédemment : PEAR::DB, Snoopy, TreeHierarchy, phpmailer.

B) Organisation des fichiers et répertoires

Lorsque l'on regarde dans le dossier où se trouvent les fichiers de VCOOP (/home/web/vcoop.net par exemple), on trouve les sous-dossiers suivants :

- **admin/** : contient les scripts spécifiques à l'administration.
- **cache/** : contient les informations en cache (gabarits smarty compilés, copie de travail des flux RSS).
- **html/** : contient les pages HTML, les scripts PHP (hors définitions des fonctions et classes) et les images.
- **includes/** : contient l'ensemble des fichiers «includes», ceux provenant des classes externes, comme ceux de l'application VCOOP en elle même (classes, définitions des fonctions).
- **scripts/** : contient quelques scripts nécessaires au bon fonctionnement de VCOOP (mise à jour des sources, indexation du moteur de recherche, ...)
- **sql/** : contient les scripts SQL permettant de créer la structure de la base.
- **templates/** : contient les gabarits (Smarty) des pages.
- **tmp/** : dossier pour recevoir d'éventuels fichiers temporaires.

1) Scripts d'administration

Le dossier **admin/** contient quelques scripts d'administration en ligne. Ces scripts sont relativement peu nombreux, les pages «publiques» s'adaptant généralement fonction de l'utilisateur connecté.

Ce dossier contient en particulier le sous-dossier **users/** qui contient toutes les fonctions d'administration des utilisateurs de VCOOP.

2) Dossiers « html »

Le dossier **html/** contient les pages et scripts de l'interface publique de VCOOP. Différents sous-dossiers regroupent les différents groupes de fonctionnalité de l'application.

- **alerte** : fonctionnalités concernant la gestion des alertes.
- **articles** : fonctionnalités les articles (affichage).
- **newsletters** : fonctionnalités concernant la gestion des newsletters.
- **forum et forums** : interface avec le forum (phpBB).
- **rubriques** : fonctionnalités concernant la gestion des rubriques.
- **sources** : fonctionnalités concernant la gestion des sources (internes et externes).
- **users** : fonctionnalités concernant les autres utilisateurs (affichage du profil, modification de son propre profil).
- **wip** : Work In Progress (mes articles, mes sources, ...).

Viennent ensuite des sous dossiers contenant des données statiques (images, contenus) ou les fichiers des classes Javascript utilisées :

- `htmlarea` : fichiers de l'éditeur Wysiwyg HtmlArea.
- `images` : images (charte graphique, icones, ...).
- `jscooktree` : fichiers de la classe d'affichage d'arbre JavaScript.
- `static` : pages de contenu statiques (aide, ...).

Sont également présents des sous dossiers à usage essentiellement interne à VCOOP :

- `data` : données des newsletters des utilisateurs.
- `indexer` : fichiers spécifiques accédés par Mnogosearch pour effectuer l'indexation des articles (accès interne).
- `plugins` : plugins pour la transformation de sources spécifiques en RSS (accès interne).

Les autres fichiers contenus dans le dossier `html/` concernent essentiellement des fonctionnalités qui n'avaient pas vraiment leur place dans les sous-dossiers décrits précédemment. Parmi ceux ci, on notera plus particulièrement les fichiers suivants :

- `affiche_article` : affichage d'un article.
- `ajoute_commentaire`, `supprime_commentaire.php` : gestion des commentaires des articles.
- `conf.js.php` : passerelle entre les scripts PHP et les scripts JavaScript pour transmettre la valeur des variables communes.
- `conf.php` : contient la configuration globale de VCOOP (adresse mail de l'administrateur, valeur des "meta keywords", paramètres de la base de données, nom des dossiers utilisés, ...).
- `contact.html`, `contact_send.php` : page de contact.
- `envoi_ami.php` : pour l'envoi d'un article, d'une source, ... à un ami.
- `index.php` : page d'accueil.
- `liste_articles.php` : liste des derniers articles.
- `liste_selection`, `ajoute_selection.php`, `selection.php`, `sel_script.js` : gestion de la sélection d'articles personnelle.
- `redirect.php`, `redirect_top.html` : scripts de redirection vers un article externe à VCOOP.
- `search.php` : affichage des résultats du moteur de recherche.
- `style.css.php`, `style_edition.css.php` : feuilles de style, générées en partie de façon dynamique.

3) Dossiers « includes »

On trouve dans le dossier `includes/` les fichiers constituant les classes internes de VCOOP ainsi que des fichiers provenant de classes externes.

Dans un premier temps, intéressons nous aux classes externes dont dépendent certaines fonctionnalités de VCOOP.

- PEAR¹ : les fichiers de PEAR doivent être placés dans le dossier `includes/PEAR/`
- PEAR::DB² : les fichiers de PEAR::DB doivent être placés dans le dossier `includes/PEAR`
- LiveUser³ : les fichiers de LiveUsers doivent être placés dans le dossier `includes/PEAR/LiveUser`
- ADODB⁴ : les fichiers d'ADODB (version testée : 1.5.0rc2) doivent être placés dans le dossier `includes/adodb/`
- FeedCreator⁵ : les fichiers de la classe FeedCreator (version testée : 1.5) doivent être placés dans le dossier `includes/feedcreator/`
- MagpieRSS⁶ : les fichiers de la classe MagpieRSS (version testée : 0.52 et 0.6a pour le support Atom) doivent être placés dans le dossier `includes/magpierss/`
Attention, MagpieRSS inclue sa propre copie de Snoopy, qui est également utilisée par VCOOP. Il est donc nécessaire de modifier le fichier `includes/magpierss/extlib/Snoopy.class.inc` et de rajouter au début une ligne contenant :

```
if (class_exists("Snoopy"))      return;
```
- mimemessage⁷ : les fichiers de la classe mimemessage doivent être placés dans le dossier `includes/mimemessage/`
- PHPMailer⁸ : les fichiers de PHPMailer doivent être placés dans le dossier `includes/phpmailer/`
- Smarty⁹ : les fichiers de Smarty (version testée : 1.71) doivent être placés dans le dossier `includes/smarty/`
- Snoopy¹⁰ : les fichiers de la classe Snoopy (version testée : 1.01) doivent être placés dans le dossier `includes/snoopy/`
Attention, MagpieRSS inclue sa propre copie de Snoopy, qui est également utilisée par VCOOP. Il est donc nécessaire de modifier le fichier `includes/magpierss/extlib/Snoopy.class.inc` et de rajouter au début une ligne contenant :

```
if (class_exists("Snoopy"))      return;
```

1 PEAR : <http://pear.php.net/packages/PEAR>

2 PEAR::DB : <http://pear.php.net/package/DB>

3 LiveUser : <http://pear.php.net/package/LiveUser> et une copie patchée pour le fonctionnement avec VCOOP : <http://vcoop.net/tarballs/LiveUser-vcoop.tgz>

4 ADODB : <http://adodb.sourceforge.net/>

5 FeedCreator : <http://www.bitfolge.de/rsscreator>

6 MagpieRSS : <http://magpierss.sourceforge.net/>

7 mimemessage : <http://www.phpclasses.org/mimemessage>

8 PHPMailer : <http://phpmailer.sourceforge.net/>

9 Smarty : <http://smarty.php.net>

10 Snoopy : <http://snoopy.sourceforge.net/>

- PHP-TreeHierarchy¹¹ : les fichiers de la classe PHP-TreeHierarchy doivent être placés dans le dossier `includes/tree-hierarchy/`
- PhpBB¹² : les fichiers de PhpBB doivent être copiés dans le dossier `html/forums/`
Il faut également installer le plugin `phpbb_fetch_all`¹³ dans le dossier `html/forums/mods/phpbb_fetch_all/`

Les autres dossiers et fichiers sont spécifiques à VCOOP :

- `display.php` : contient des fonctions d'affichage génériques
- `include_all.php` : contient tous les appels «include» des fichiers nécessaires
- `smarty_plugins/` : contient des plugins Smarty développés pour VCOOP
- `vcoop/` : contient l'ensemble des classes de VCOOP. Le contenu de ce dossier sera décrit plus en détail par la suite.

4) Classes utilisées dans VCOOP

Dans le dossier `includes/vcoop/` sont situées les classes développées spécifiquement pour VCOOP. Les principales classes sont les suivantes :

Classe de base

Les classes développées pour VCOOP reposent toutes sur une classe de base, `vcoop_object`, qui définit les comportements communs, tels que le transfert entre le contenu de la base SQL et les propriétés de l'objet, l'accès aux champs communs à tous les types d'objets, etc.

Articles (items)

La classe `vcoop_item` définit le comportement des éléments de contenus de VCOOP. Elle se décline en deux sous-classes, selon que les contenus sont produits dans `vcoop` (`vcoop_item_internal`) ou proviennent de l'extérieur (`vcoop_item_rss`).

Les articles externes sont identifiés par le lien externe permettant de consulter l'article complet. Ce lien sert de base pour éviter d'insérer en doublon un même contenu. La description de l'article fournie par la source externe est stockée.

Un article interne comporte un résumé, qui utilise le même champ que la description d'un article externe. Pour un article plus long sera composé de un ou plusieurs paragraphes supplémentaires correspondant au contenu. L'équivalent du lien externe permet alors de visualiser l'article complet.

11 PHP-TreeHierarchy : <http://vcoop.net/tarballs/TreeHierarchy-0.6.0b.tar.gz> (le projet semble ne plus exister ailleurs)

12 PhpBB : <http://www.phpbb.com/>

13 `phpbbfetchall` : <http://www.phpbbfetchall.com/>

Sources

Chaque article, qu'il soit interne ou externe est relié à une source, représentée par un objet de la classe `vcoop_source`. On retrouve la même déclinaison que pour les articles avec les sources internes à VCOOP (`vcoop_source_internal`) et les sources externes (`vcoop_source_rss`).

Dans le cas d'une source externe, on trouve notamment des informations permettant d'identifier le site d'origine, la fréquence de mise à jour, etc.

La fréquence de mise à jour des sources se base d'une part sur l'information donnée par la source (sous forme d'un temps minimum à respecter entre deux interrogations), et d'autre part sur un calcul statistique par rapport à la fréquence mesurée d'entrée de nouveaux articles, afin de limiter les traitements inutiles.

Lors de l'ajout d'une source externe, une vérification est effectuée par rapport aux sources et articles déjà référencés pour essayer de détecter les risques de doublons. Dans le cas où un risque de doublon est détecté, un message avertissant l'utilisateur est affiché.

Alertes

Les alertes (classe `vcoop_alerte` et `vcoop_alerte_item`) sont des « recherches enregistrées » exécutées lorsque de nouveaux contenus sont introduits dans VCOOP, permettant aux utilisateurs enregistrés d'être tenus au courant de l'apparition de contenus en rapport avec leurs centres d'intérêt.

Newsletters et éditions

Les utilisateurs ayant un niveau d'accès suffisant ont la possibilité de rassembler plusieurs articles qu'ils ont écrits ou sélectionnés sous forme d'éditions de leur newsletter. La classe `vcoop_newsletter` définit les éléments communs, tel que le nom de la newsletter, les abonnés, ..., alors que la classe `vcoop_edition` rassemble les éléments spécifiques d'une édition donnée.

Une édition est composée de un ou plusieurs segments. Les paragraphes correspondent soit à des titres, soit à des articles issus de votre sélection, soit à des articles que vous avez rédigés.

Il est possible de choisir le gabarit d'affichage d'une newsletter parmi ceux qui sont prédéfinis. Les éditions héritent du choix par défaut de la newsletter lors de leur création, mais il est également possible d'en choisir un autre ultérieurement.

La structure par défaut (nombre de parties et titres) d'une newsletter peut-être prédéfinie. Les éditions en héritent lors de leur création.

Utilisateurs et droits d'accès

La gestion des utilisateurs et des droits d'accès reposent sur la classe `LiveUser` encapsulée dans la classe `vcoop_user`, permettant de gérer tous les aspects de la gestion des droits qui sont spécifiques à VCOOP.

Autres fonctions et classes

Parmi les autres fonctions, on peut notamment répertorier celles qui servent à la collecte des commentaires sur les articles et les sources, les fonctions de recherche (s'appuyant, pour la recherche plein texte sur MnegoSearch), l'interface avec Sympa (pour les envois de mails pour les newsletters) ou PhpBB (pour gérer les forums) et les fonctions d'envoi de mails.

5) Gabarits

L'ensemble des gabarits Smarty utilisés sont regroupés dans le dossier `templates/`. Les fichiers dont le nom commence par `intro_` correspondent généralement aux petits textes d'introduction affichés dans les pages de VCOOP, par exemple `intro_liste_sources.smarty` permet d'indiquer le nombre de sources référencées dans VCOOP.

Le sous-dossier `templates/static/` contient l'ensemble des gabarits utilisés dans les pages statiques (aide, présentation, etc.). Celles ci sont donc composées d'un fichier dans `html/static/` et d'un gabarit dans `templates/static/`.

D'une manière générale, l'utilisation des gabarits Smarty fait un usage assez intensif des inclusions. Parmi les gabarits génériques, on peut en particulier noter :

- `titre.smarty` : ce gabarit est utilisé pour construire le titre des pages, avec la possibilité de spécifier un titre (`$page.titre`), un sous-titre (`$page.sous_titre` et `$page.url_sous_titre`), une liste d'actions (`$page.actions`) et le nom d'une page d'aide (`$page.aide`).
- `boite_titre_smarty` : ce gabarit permet de construire les boîtes affichées à l'intérieur des pages de contenus. On peut lui spécifier un titre (`$titre` et `$url_titre`) et un sous-titre (`$sous_titre` et `$url_sous_titre`).
- `boite_titre_col.smarty` : ce gabarit permet de générer une boîte comme celles qui sont affichées dans les colonnes. On lui transmet un titre (`$titre` et `$url_titre`).
- `top.smarty` : ce gabarit permet d'afficher les boîtes « les derniers xxx ». On lui transmet le type (`$type`) et les données (`$info`). Ce gabarit utilise `boite_titre_col.smarty`.
- `editeur_html.smarty` : ce gabarit permet de mettre en place un éditeur Wysiwyg avec HtmlArea. Il faut lui transmettre le nom du champ (`$field_name`) et sa valeur (`$field_value`) actuelle. Il est également possible d'influer sur ses dimensions (largeur (`$cols`) et hauteur (`$rows`)).
- `pager.smarty` : ce gabarit permet de générer la barre de navigation dans les résultats d'une recherche. Il est possible de passer un message à afficher tel quel (`$pager_message`), sinon il faut transmettre le nombre total de résultats (`$search.nb_total`) et le nombre affiché (`$search.nb_select`). Ce gabarit utilise le plugin Smarty spécifique `vcoop_pager`.

- `search_result.smarty` : ce gabarit permet d'afficher les résultats d'une recherche. Il utilise le gabarit `article.smarty` pour afficher un article ou `source.smarty` pour afficher une source. Les données lui sont transmises dans un tableau (`$data`).

On trouve ensuite plusieurs gabarits qui servent pour construire l'entête et le pied de page.

Pour l'entête : `header.smarty` qui fait appel à `header_init.smarty`, `logo.smarty`, `menu.smarty` et `debut_contenu.smarty`

Pour le pied de page : `footer.smarty` qui fait appel à `fin_contenu.smarty` et `signature.smarty`

Les gabarits `source.smarty` et `article.smarty` (et dans une moindre mesure `newsletter.smarty`) sont utilisés pour tous les affichages du type d'objet correspondant. Par conséquent, ils contiennent de nombreux tests qui font varier l'affichage en fonction du contexte.

C) Procédure d'installation

Pour la mise en place des sources, il est nécessaire de disposer d'une installation d'Apache avec PHP. Il est également nécessaire de disposer d'une installation de MngoSearch et Sympa.

Pour PHP, les options suivantes doivent être activées :

- XML expat (pour le fonctionnement de MagpieRSS)
- iconv ou recode (pour gérer les conversions de jeu de caractères)
- mysql
- mnogosearch (il faut une version de l'extension supérieure à 1.86, car elle doit intégrer une correction d'un bug découvert pendant le développement de VCOOP, c'est normalement le cas des version de PHP actuelles)
- locales françaises (pour pouvoir formater correctement les dates en français)

1) Mise en place des scripts PHP

Nous considérerons à partir d'ici que l'installation sera effectuée dans un dossier nommé `/home/web/vcoop.net` avec comme nom de domaine `vcoop.net`.

Il est nécessaire de disposer de deux bases Mysql, l'une pour les données de VCOOP (sources, articles, ...), la deuxième pour le stockage des informations de MngoSearch.

Les classes externes à VCOOP listées en début de document doivent être mises en place dans `/home/web/vcoop.net/includes` tel que décrit à la page 6.

2) Contenu de la base de données

En se connectant sur la base de données (avec PHPMyAdmin, par exemple), importez le contenu des fichiers :

- `sql/vcoop_tables.sql`
- `sql/vcoop_threads.sql`
- `sql/vcoop_liveuser.sql`
- `sql/vcoop_phpbb.sql`

3) Configuration de VCOOP

Pour la configuration, il est nécessaire de modifier le fichier `html/conf.php` pour l'adapter aux paramètres de l'installation en cours.

Parmi les variables à renseigner, on trouve :

- `$dsn` : paramètres de la base Mysql principale
- `$dsn_udm` : paramètres de la base Mysql pour MngoSearch
- `VCOOP_HTTP_ROOT` : racine du site (<http://vcoop.net/> par exemple)
- `VCOOP_ADM_ROOT` : racine des pages d'administration (<http://vcoop.net/admin/> par exemple)

- VCOOP_INDEXER_ROOT : racine des liens tels que vus par l'indexeur de MnoGoSearch (généralement identique à VCOOP_HTTP_ROOT)
- VCOOP_CACHE_DIR : dossier dans lequel stocker les fichiers du cache
- \$VCOOP_CACHE : activer (true) ou non (false) le cache
- VCOOP_SMTP_HOST : nom du serveur SMTP à utiliser pour les envois de mails
- VCOOP_EMAIL_ADMIN : adresse mail de l'administrateur
- VCOOP_EMAIL_FROM : adresse mail utilisée pour l'expédition des mails
- VCOOP_EMAIL_FROM_NAME : nom utilisé pour l'expédition des mails
- VCOOP_SYMPA_LISTS_DIR : dossier racine des listes gérées par Sympa
- VCOOP_SYMPA_GROUP : groupe Unix sous lequel fonctionne Sympa
- VCOOP_SYMPA_CHMOD : droits à appliquer sur les fichiers de Sympa
- VCOOP_SYMPA_CHMOD_DIR : droits à appliquer sur les dossiers de Sympa
- VCOOP_SYMPA_ALIAS : nom du fichier contenant les alias mail que Sympa va gérer
- VCOOP_SYMPA_NEWALIASES : emplacement de la commande à lancer pour mettre à jour la base des alias après une modification

4) Configuration de PhpBB

Il est nécessaire de créer le fichier config.php de PhpBB (html/forums/config.php) sur la base des lignes suivantes :

```
<?php
$dbms = 'mysql';
$dbhost = 'localhost'; ## nom du serveur mysql
$dbname = 'vcoop';      ## nom de la base mysql
$dbuser = 'vcoop';     ## nom d'utilisateur
$dbpasswd = 'vcoop';   ## mot de passe
$table_prefix = 'phpbb_';
define('PHPBB_INSTALLED', true);
?>
```

5) Configuration d'Apache

Certaines fonctions nécessitant d'écrire dans l'espace disque où sont stockés les pages HTML, il est conseillé de faire fonctionner PHP avec l'option «Safe Mode» désactivée. Dans le cas où elle serait activée, certaines fonctionnalités concernant les newsletters et la gestion des abonnés risquent de ne pas fonctionner correctement ou vont nécessiter une adaptation.

Pour assurer une coopération correcte entre le serveur web et Sympa, il sera probablement nécessaire de mettre l'utilisateur sous lequel fonctionne le serveur web (généralement www ou www-data) dans le groupe de Sympa (généralement sympas). Par exemple :

```
$ id sympy
uid=104(sympy) gid=105(sympy) groups=105(sympy)
$ id www-data
uid=33(www-data) gid=33(www-data) groups=33(www-data),105(sympy)
```

Le sous-dossier html/data doit également avoir des droits spécifiques permettant à l'utilisateur sous lequel fonctionne le serveur Web d'écrire dedans. Par exemple :

```
drwxrwxrwx  20 vcoop  vcoop          4096 Oct 25 16:03 html/data/
```

La configuration du virtualhost d'Apache ne nécessite rien de très spécifique, l'exemple suivant suffit, mais peut être adapté :

```
<VirtualHost *>
  ServerAlias vcoop.net
  ServerAlias www.vcoop.net
  DocumentRoot /home/web/vcoop.net/public_html
  TransferLog /var/log/httpd/vcoop.net
  Alias /admin /home/web/vcoop.net/admin
  AddType application/x-httpd-php .php .inc .phtml .htm .html
  DirectoryIndex index.html index.php index.htm index.phtml
  <Directory /home/web/vcoop.net>
    AllowOverride All
    Options -Indexes
  </Directory>
</VirtualHost>
```

Pour les plugins RSS pour les sources externes, un deuxième virtualhost peut être mis en place (pour l'autorisation, remplacer 1.2.3.4 par l'adresse ip du serveur sur lequel l'installation est effectuée :

```
<VirtualHost *>
  ServerName plugins.vcoop.net
  DocumentRoot /home/web/plugins.vcoop.net/public_html
  TransferLog /var/log/httpd/plugins.vcoop.net
  AddType application/x-httpd-php .php .inc .phtml .htm .html
  DirectoryIndex index.html index.php index.htm index.phtml
  <Directory /home/web/plugins.vcoop.net>
    AllowOverride All
    Options -Indexes
  </Directory>
  <Location />
    Order deny,allow
    Deny from all
    Allow from 1.2.3.4
    Allow from 127.0.0.1
  </Location>
</VirtualHost>
```

6) Configuration de MnogoSearch

Il est nécessaire de compiler MnogoSearch avec le support Mysql, par exemple avec une commande comme celle ci :

```
./configure --with-mysql --disable-syslog --prefix=/opt/mnogosearch-3.2
```

Une fois installé (ici dans `/opt/mnogosearch-3.2`), il suffit alors de copier les fichiers `scripts/common.conf` et `scripts/vcoop.conf` dans `/opt/mnogosearch-3.2/etc/`

Modifier le fichier `vcoop.conf` pour tenir compte des paramètres de votre base de données.

Avec une version récente de MngoSearch, il suffit normalement d'exécuter la commande suivante :

```
/opt/mnogosearch-3.2/etc $ ../sbin/indexer vcoop.conf -Ecreate
```

qui devrait créer toute la structure des tables nécessaires au bon fonctionnement de MngoSearch.

7) Scripts périodiques

Le dossier `scripts/` contient des scripts écrits en shell qui permettent la mise à jour des sources et l'indexation. Il peut être nécessaire de les modifier pour correspondre à la configuration précise du serveur sur lequel se fait l'installation.

Le script `upd_vcoop.sh` doit être lancé périodiquement (environ une fois par minute) pour récupérer les derniers articles des sources. Il peut être planifié à l'aide de Cron. Le script `boucle_vcoop.sh` peut également être utilisé pour le lancer «en boucle» depuis un terminal.

Le script `upd_vcoop_alertes.sh` doit également être lancé régulièrement, afin d'indexer les nouveaux contenus et de générer les résultats d'alertes correspondants. Il peut être planifié à l'aide de Cron, par exemple avec un lancement tous les quarts d'heure.

D) Description des tables spécifiques à VCOOP

Parmi les données stockées dans la base Mysql, un certain nombre sont spécifiques aux outils utilisés (PhpBB, LiveUser, PHP-TreeHierarchy) et ne seront pas détaillées.

1) TABLE source

Nom du champ	Type	Description
id_sou	INTEGER NOT NULL	Identifiant unique
id_owm	VARCHAR(32) NOT NULL	Id créateur de la source
id_lic	INTEGER NOT NULL	Identifiant licence
date_create	DATETIME	Date création
date_update	DATETIME	Date dernière modification
status	INTEGER	Etat
id_rubg	INTEGER	Id classement global
id_rubo	INTEGER	Id classement créateur
views	INTEGER	Nombre d'affichages
what	INTEGER	RSS, Blog, ...
title	VARCHAR(255)	Nom de la source
link	VARCHAR(255)	Lien pour charger le contenu
description	TEXT	Description de la source
access_cond	VARCHAR(255)	Conditions d'accès (payant, enregistrement, ...)
language	VARCHAR(255)	Langue
license	VARCHAR(255)	Licence
license_logo	VARCHAR(255)	Url logo pour signaler la licence
license_url	VARCHAR(255)	Url vers texte de la licence
internal	INTEGER	Source interne à VCOOP ?
home_link	VARCHAR(255)	Adresse du site "parent"
copyright	VARCHAR(255)	Information sur le copyright originel
author	VARCHAR(255)	Auteur
logo_url	VARCHAR(255)	URL image du logo
logo_alt	VARCHAR(255)	Texte "alt" du logo
logo_wid	VARCHAR(255)	Texte "alt" du logo
logo_hei	VARCHAR(255)	Texte "alt" du logo

Nom du champ	Type	Description
last_fetch	DATETIME	Dernier chargement (externe)
refresh	INTEGER	Fréquence de rafraîchissement en minutes
nb_items	INTEGER	nb d'articles dans cette source
nb_commentaires	INTEGER	nb de commentaires

2) TABLE item

Nom du champ	Type	Description
id_ite	INTEGER NOT NULL	Identifiant unique
id_typ	INTEGER NOT NULL	Type de document
id_sou	INTEGER NOT NULL	Identifiant de la source
id_own	VARCHAR(32) NOT NULL	Id créateur de l'article
id_lic	INTEGER NOT NULL	Identifiant licence
date_create	DATETIME	Date de création
date_update	DATETIME	Date de dernière modification
status	INTEGER	Etat
id_art	INTEGER	Identifiant article si interne à vcoop
views	INTEGER	Nombre d'affichages
title	VARCHAR(255)	Titre
link	VARCHAR(255)	Lien vers l'article
description	TEXT	Description de l'article
comments_url	VARCHAR(255)	URL vers pages avec des commentaires sur l'article
language	VARCHAR(255)	Langue
copyright	VARCHAR(255)	Information sur le copyright originel
author	VARCHAR(255)	Auteur
guid	VARCHAR(255)	Chaîne d'identification de l'article
nb_commentaires	INTEGER	nb de commentaires de cet article
indexe	INTEGER	article a indexer ?
specific_themes	INTEGER	thèmes spécifiques (1) ou hérités de la source (0)

3) TABLE selection

Nom du champ	Type	Description
id_sel	INTEGER NOT NULL	Identifiant unique
id_own	VARCHAR(32) NOT NULL	Identifiant utilisateur
id_ite	INTEGER NOT NULL	Identifiant item
date_create	DATETIME	Date ajout (pour le tri)

4) TABLE commentaire

Commentaires sur les items.

Nom du champ	Type	Description
id_com	INTEGER NOT NULL	Identifiant unique
id_own	VARCHAR(32) NOT NULL	Identifiant utilisateur
id_ite	INTEGER NOT NULL	Identifiant item
parent	INTEGER NOT NULL	id commentaire "parent"
date_create	DATETIME	Date ajout (pour le tri)
subject	VARCHAR(255)	titre
content	TEXT	texte du commentaire
notation	INTEGER	Évaluation de l'article

5) TABLE scommentaire

Commentaires sur les sources.

Nom du champ	Type	Description
id_sco	INTEGER NOT NULL	Identifiant unique
id_own	VARCHAR(32) NOT NULL	Identifiant utilisateur
id_sou	INTEGER NOT NULL	Identifiant source
parent	INTEGER NOT NULL	id commentaire "parent"
date_create	DATETIME	Date ajout (pour le tri)
subject	VARCHAR(255)	titre
content	TEXT	texte du commentaire
notation	INTEGER	Évaluation de l'article

6) TABLE item_rubrique

Lien entre un article et un thème.

Nom du champ	Type	Description
id_ite	INTEGER NOT NULL	Identifiant item
id_nod	INTEGER NOT NULL	Identifiant rubrique

7) TABLE source_rubrique

Lien entre une source et un thème

Nom du champ	Type	Description
id_sou	INTEGER NOT NULL	Identifiant source
id_nod	INTEGER NOT NULL	Identifiant rubrique
id_own	VARCHAR(32) NOT NULL	Identifiant utilisateur

8) TABLE profil

Informations personnelles sur un utilisateur.

Nom du champ	Type	Description
id_profil	VARCHAR(32) NOT NULL	Identifiant unique
date_create	DATETIME NOT NULL	Date de création
date_update	DATETIME NOT NULL	Date de m.à.j
nom	VARCHAR(40) NOT NULL	Nom
prenom	VARCHAR(40) NOT NULL	Prénom
email	VARCHAR(40) NOT NULL	Adresse mail
adresse	VARCHAR(255) NOT NULL	Adresse postale
cp	VARCHAR(5) NOT NULL	Code postal
ville	VARCHAR(40) NOT NULL	Ville
cv	TEXT NOT NULL	Texte du "CV"
interet	TEXT NOT NULL	Centres d'intérêts
id_thr	INTEGER	Identifiant arbre des thèmes
profession	VARCHAR(255) NOT NULL	Profession
organisation	VARCHAR(255) NOT NULL	Organisation
url_organisation	VARCHAR(255) NOT NULL	URL de l'organisation
autres_sites	VARCHAR(255) NOT NULL	
vu_coord_perso	CHAR(1) NOT NULL	Flag pour visibilité des coordonnées personnelles
vu_coord_pro	CHAR(1) NOT NULL	... visibilité informations pro.
vu_alertes	CHAR(1) NOT NULL	... visibilité alertes

Nom du champ	Type	Description
<code>vu_sources</code>	<code>CHAR(1) NOT NULL</code>	... visibilité des sources
<code>vu_wip</code>	<code>CHAR(1) NOT NULL</code>	... visibilité du work in progress
<code>vu_proches</code>	<code>CHAR(1) NOT NULL</code>	... visibilité des proches
<code>vu_forums</code>	<code>CHAR(1) NOT NULL</code>	... visibilité des forums
<code>vu_articles</code>	<code>CHAR(1) NOT NULL</code>	... visibilité des articles
<code>vu_selection</code>	<code>CHAR(1) NOT NULL</code>	... visibilité de la sélection

9) TABLE paragraphe

Paragraphe (élément d'un article "long")

Nom du champ	Type	Description
<code>id_par</code>	<code>INTEGER NOT NULL</code>	Identifiant unique
<code>id_typ</code>	<code>INTEGER NOT NULL</code>	Type de document
<code>id_ite</code>	<code>INTEGER NOT NULL</code>	Identifiant de l'item
<code>title</code>	<code>VARCHAR(255)</code>	Titre
<code>content</code>	<code>TEXT</code>	Contenu
<code>image</code>	<code>VARCHAR(255)</code>	Image
<code>link</code>	<code>VARCHAR(255)</code>	Lien
<code>rank</code>	<code>INTEGER</code>	Ordre dans l'article

10) TABLE typedocs

Type de documents (plus utilisé)

Nom du champ	Type	Description
<code>id_typ</code>	<code>INTEGER NOT NULL</code>	Identifiant unique
<code>name</code>	<code>VARCHAR(255)</code>	Nom
<code>plural</code>	<code>VARCHAR(255)</code>	Pluriel

11) TABLE newsletter

Nom du champ	Type	Description
<code>id_new</code>	<code>INTEGER NOT NULL</code>	Identifiant unique
<code>status</code>	<code>INTEGER</code>	Etat
<code>id_own</code>	<code>VARCHAR(32) NOT NULL</code>	Identifiant utilisateur
<code>date_create</code>	<code>DATETIME NOT NULL</code>	Date de création
<code>title</code>	<code>VARCHAR(255)</code>	Titre

Nom du champ	Type	Description
<code>gabarit</code>	<code>VARCHAR(255)</code>	chemin du gabarit
<code>entete</code>	<code>TEXT</code>	entête de newsletter
<code>use_entete</code>	<code>INTEGER</code>	utiliser l'entête
<code>description</code>	<code>TEXT</code>	Description
<code>pied</code>	<code>TEXT</code>	pied de la newsletter
<code>use_pied</code>	<code>INTEGER</code>	utiliser le pied
<code>use_sommaire</code>	<code>INTEGER</code>	utiliser le sommaire automatique
<code>link</code>	<code>VARCHAR(255)</code>	Lien
<code>language</code>	<code>VARCHAR(255)</code>	Langue

12) TABLE npartie

Parties d'une newsletter.

Nom du champ	Type	Description
<code>id_npa</code>	<code>INTEGER NOT NULL</code>	Identifiant unique
<code>id_new</code>	<code>INTEGER NOT NULL</code>	Identifiant newsletter
<code>title</code>	<code>VARCHAR(255)</code>	Titre
<code>rank</code>	<code>INTEGER</code>	ordre

13) TABLE edition

Nom du champ	Type	Description
<code>id_edi</code>	<code>INTEGER NOT NULL</code>	Identifiant unique
<code>id_new</code>	<code>VARCHAR(32) NOT NULL</code>	Identifiant newsletter
<code>status</code>	<code>INTEGER</code>	Etat
<code>date_create</code>	<code>DATETIME NOT NULL</code>	Date de création
<code>date_update</code>	<code>DATETIME NOT NULL</code>	Date de modification
<code>date_publish</code>	<code>DATETIME NOT NULL</code>	Date de publication
<code>gabarit</code>	<code>VARCHAR(255)</code>	chemin du gabarit
<code>title</code>	<code>VARCHAR(255)</code>	Titre
<code>entete</code>	<code>TEXT</code>	entête de l'édition
<code>use_entete</code>	<code>INTEGER</code>	utiliser l'entête
<code>description</code>	<code>TEXT</code>	Description
<code>pied</code>	<code>TEXT</code>	pied de l'édition

Nom du champ	Type	Description
<code>use_pied</code>	INTEGER	utiliser le pied
<code>use_sommaire</code>	INTEGER	utiliser le sommaire automatique
<code>link</code>	VARCHAR(255)	Lien

14) TABLE segment

Segment d'une édition.

Nom du champ	Type	Description
<code>id_seg</code>	INTEGER NOT NULL	Identifiant unique
<code>id_parent</code>	INTEGER NOT NULL	Identifiant parent
<code>id_typ</code>	INTEGER NOT NULL	Type de document
<code>id_edi</code>	VARCHAR(32) NOT NULL	Identifiant edition
<code>title</code>	VARCHAR(255)	Titre
<code>description</code>	TEXT	Description
<code>id_ite</code>	INTEGER NOT NULL	Identifiant item
<code>rank</code>	INTEGER	ordre

15) TABLE newsletter_pending_subscribe

Abonnés d'une newsletter en attente de validation.

Nom du champ	Type	Description
<code>id_sub</code>	VARCHAR(50) NOT NULL	Identifiant unique
<code>id_new</code>	INTEGER NOT NULL	Identifiant newsletter
<code>insc_date</code>	DATETIME NOT NULL	Date inscription
<code>email</code>	VARCHAR(255)	Adresse email
<code>action</code>	VARCHAR(20)	Action (subscribe, signoff)

16) TABLE newsletter_subscribe

Abonnés d'une newsletter.

Nom du champ	Type	Description
<code>id_sub</code>	VARCHAR(50) NOT NULL	Identifiant unique
<code>id_new</code>	INTEGER NOT NULL	Identifiant newsletter
<code>insc_date</code>	DATETIME NOT NULL	Date inscription
<code>email</code>	VARCHAR(255)	Adresse email

17) TABLE alerte_item

Résultats d'alertes.

Nom du champ	Type	Description
id_alerte_item	INTEGER NOT NULL	Identifiant unique
type	INTEGER	type article, source, forum ...
id_contenu	VARCHAR(32) NOT NULL	Identifiant edition
id_owner	VARCHAR(32)	Propriétaire
id_alerte	INTEGER	alerte qui a généré
vu	CHAR(1)	Vu ou non
date_create	DATETIME	date de création de l'alerte
language	VARCHAR(255) NOT NULL	Langue

18) TABLE alerte

Nom du champ	Type	Description
id_alerte	INTEGER NOT NULL	Identifiant unique
id_owner	VARCHAR(32) NOT NULL	Identifiant utilisateur
date_create	DATETIME	date de création de l'alerte
keyword	VARCHAR(255)	Mots clés
searchtype	INTEGER	type article, source, forum
selection	VARCHAR(255) NOT NULL	Identifiant objet (dépend du type)
actif	CHAR(1)	status
last_fetch	DATETIME NOT NULL	Dernière m.à.j
nb_items	INTEGER NOT NULL	Nom d'articles sélectionnés
lang	CHAR(2) NOT NULL	Langue
word_rech	INTEGER NOT NULL	

19) TABLE forum_group_owner

Lien entre un forum phpbb et le propriétaire du forum.

Nom du champ	Type	Description
group_id	INTEGER NOT NULL	Identifiant du forum
auth_user_id	VARCHAR(32) NOT NULL	Identifiant utilisateur