

# Manuel d'exploitation de l'application Vitamin version 1.0



Ce document a pour but de décrire les procédures  
d'exploitation de l'application Vitamin

# 1 - Gestion du document

---

## 1.1 Auteurs

	Nom	Société	Date
Rédigé par	Thierry Martins	Asyrès	06/02/2006
	Ronan Balaven	Asyrès	7/02/2006
	Thierry Aimé	DGME/SDAE	04/06/2006

## 1.2 Gestion de version

Version	Date	Description	Editeurs
1.0	30/005/2006	Version accompagnant la version 1.0 de Vitamin	

## 1.3 Documents de référence

N° Référence	Document	Référence

## 1.4 Liens

Liens amont	/
	/
Liens aval	<a href="http://www.python.org">Http://www.python.org</a>
	<a href="http://www.zope.org">http://www.zope.org</a>
	<a href="http://www.cps-project.org">http://www.cps-project.org</a>

## 1.5 Licence

Ce document est publié sous licence GPL, comme l'ensemble des codes sources composant le programme Vitamin.

Copyright 2006 DGME

This file is part of Vitamin.

Vitamin is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

Vitamin is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with Vitamin; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

# 2 - Sommaire

---

<b>1 - GESTION DU DOCUMENT.....</b>	<b>2</b>
<b>1.1 Auteurs .....</b>	<b>2</b>
<b>1.2 Gestion de version.....</b>	<b>2</b>
<b>1.3 Documents de référence.....</b>	<b>2</b>
<b>1.4 Liens.....</b>	<b>2</b>
<b>1.5 Licence.....</b>	<b>3</b>
<b>2 - SOMMAIRE.....</b>	<b>4</b>
<b>3 - INTRODUCTION .....</b>	<b>6</b>
<b>4 - PRINCIPES GÉNÉRAUX ET ARCHITECTURE.....</b>	<b>7</b>
<b>4.1 Principes généraux de l'application.....</b>	<b>7</b>
<b>4.2 Architecture logique.....</b>	<b>7</b>
<b>4.3 Composants logiciels.....</b>	<b>7</b>
<b>4.4 Descriptif des flux applicatifs.....</b>	<b>7</b>
<b>5 - DÉMARRAGE ET ARRÊT DU SYSTÈME.....</b>	<b>8</b>
<b>5.1 Arrêt-relande des machines.....</b>	<b>8</b>
<b>5.2 Arrêt-relande des composants.....</b>	<b>8</b>
5.2.1 - Mode opératoire pour démarrer/arrêter manuellement un serveur http.....	8
5.2.2 - Mode opératoire pour démarrer/arrêter manuellement le serveur d'application Zope .....	8
<b>6 - ORDONNANCEMENT DES TRAITEMENTS PAR LOT.....</b>	<b>9</b>
<b>7 - SAUVEGARDES, RESTAURATIONS ET MAINTENANCES.....</b>	<b>10</b>
<b>7.1 Sauvegarde du système.....</b>	<b>10</b>
<b>7.2 Restauration du système.....</b>	<b>10</b>
<b>7.3 Maintenance générale des composants et de l'application.....</b>	<b>10</b>
<b>7.4 Page de Garde de la maintenance.....</b>	<b>10</b>
<b>8 - GESTION DES BASES DE DONNÉES.....</b>	<b>11</b>
<b>8.1 Accès à la ZODB.....</b>	<b>11</b>
<b>8.2 Gestion de données.....</b>	<b>11</b>
<b>8.3 Sauvegarde de la base de données.....</b>	<b>11</b>
<b>8.4 Autres tâches de maintenance.....</b>	<b>12</b>

<b>9 - LOCALISATION PHYSIQUE DES DONNÉES ET DES PROGRAMMES.....</b>	<b>13</b>
<b>9.1 Liste des serveurs applicatifs installés.....</b>	<b>13</b>
9.1.1 - Apache.....	13
9.1.2 - Zope.....	13
<b>9.2 Liste des composants installés pour l'application Vitamin2.....</b>	<b>13</b>
<b>10 - INTERFACES AVEC LES AUTRES SYSTÈMES.....</b>	<b>14</b>
<b>11 - SURVEILLANCE DU SYSTÈME.....</b>	<b>15</b>
<b>11.1 Surveillance de l'infrastructure.....</b>	<b>15</b>
<b>11.2 Surveillance des logiciels.....</b>	<b>15</b>
<b>11.3 Surveillance applicative (fichiers journaux).....</b>	<b>16</b>
<b>11.4 Les URL de surveillance.....</b>	<b>16</b>

# 3 - Introduction

---

Ce manuel d'exploitation de l'application Vitamin2 a pour but :

- De décrire l'architecture et les principes généraux de fonctionnement de l'application,
- De formaliser l'ensemble des opérations de démarrage, d'arrêt, de maintenance, d'administration et de support nécessaires pour exploiter l'application.

Ce manuel s'appuie sur les manuels d'administration et d'exploitation des logiciels open source utilisés ainsi que les spécificités de l'application Vitamin2 mises en place pour auditer son état.

# 4 - Principes généraux et architecture

---

## 4.1 Principes généraux de l'application

Le système Vitamin2 est un outil de travail collaboratif architecturé sur des technologies libres ZOPE CMF/CPS3. Il répond aux besoins de gestion de contenus Web, de travail collaboratif et de création et gestion de portail. Initialement développé sous le nom de Dolce puis rebaptisé Vitamin2, c'est un outil stratégique dans le développement de l'administration électronique. Développée par Asyrès, cette nouvelle version de Vitamin se veut une évolution ergonomique et fonctionnelle de l'application existante vers la plateforme CPS 3.3.8

## 4.2 Architecture logique

[A réaliser sur la base de l'architecture physique déployée pour le compte de la DGME]

## 4.3 Composants logiciels

<b>Composant logiciel</b>	<b>Description</b>	<b>Paramétrage particulier*</b>
Linux Red Hat 4 ES Enterprise	Système d'exploitation 64 bits (Evolution Pack 2 prochainement)	
Apache X.X.XX pour Linux	Serveur HTTP, développé sous licence open source par la fondation Apache	
Zope 2.8.5-final pour Linux	Serveur d'applications OpenSource programmé en python et embarquant sa propre base de données objet	
CPS 3.3.8 pour Linux	Système de gestion de contenu collaboratif basé sur le serveur d'applications Zope	

## 4.4 Descriptif des flux applicatifs

Connexions suivantes en entrée / sortie Internet :

- HTTPS

A l'intérieur de l'architecture nous dénombrons les protocoles suivants :

- HTTP : entre le serveur Web et le serveur Zope ;
- SMTP : protocole de transport des messages avec sendmail

Ces protocoles sont applicatifs et utilisent le protocole TCP/IP et UDP/IP pour les protocoles de transport.

# 5 - Démarrage et arrêt du système

---

Ce chapitre décrit les opérations à effectuer pour démarrer/arrêter le système.

## 5.1 Arrêt-reliance des machines

Mode opératoire pour démarrer les serveurs applicatifs et de données

1. Éteindre le serveur d'applications Zope :

```
$ ZOPE_HOME/bin/zopectl stop
```

2. Redémarrer le serveur :

```
$ reboot
```

3. Démarrer le serveur d'applications en tant qu'utilisateur zope :

```
$ZOPE_HOME/bin/zopectl start
```

## 5.2 Arrêt-reliance des composants

### 5.2.1 - Mode opératoire pour démarrer/arrêter manuellement un serveur http

1. Le lancement du serveur Web Apache s'effectue par la commande :

```
$ /etc/init.d/httpd start
```

2. L'arrêt du serveur Web Apache s'effectue par la commande :

```
$ /etc/init.d/httpd stop
```

### 5.2.2 - Mode opératoire pour démarrer/arrêter manuellement le serveur d'application Zope

1. Le lancement du serveur d'application Zope s'effectue par la commande :

```
$ ZOPE_HOME/bin/zopectl start
```

2. L'arrêt du serveur d'application Zope s'effectue par la commande :

```
$ ZOPE_HOME/bin/zopectl stop
```

## 6 - Ordonnancement des traitements par lot

---

Ce chapitre liste sous forme de tableau les différents traitements par lot (ou traitements batch) et indique la description, la périodicité d'exécution, les heures de début et de fin d'exécution, la durée du traitement et les contraintes d'exécution...

Tous les traitements par lots sont configurés dans le flux crontab :

Ce fichier nommé crontab se trouve dans le répertoire : /etc/

Nous distinguons deux types de traitement par lot :

- Les traitements journaliers à heure fixe ;
- Les traitements par lots à intervalles de temps (XXmn).

Le paramètre de XX minutes semble un bon compromis pour la volumétrie attendue.

Les traitements par lots quotidiens à heure fixe :

<b>Batch de suppression des comptes non validés</b>	
<b>Identifiant du traitement</b>	DELETE_USERS
<b>URL de déclenchement du batch</b>	http://<ip_server>:<port>/<objet_CPS>/scheduler_delete
<b>Libellé du traitement</b>	Batch de suppression des comptes non validés après l'inscription (pas d'inscription à une communauté, email non valide)
<b>Description du traitement</b>	Vérifie que tous les nouveaux utilisateurs dont l'inscription date de moins de 3 jours ont rejoint une communauté
<b>Périodicité</b>	Tous les jours à 01:00 AM
<b>Durée estimée</b>	Temps moyen : < 5 secondes
<b>Contraintes d'exécution</b>	Aucun
<b>Paramètre du traitement</b>	Aucun
<b>Résultat du traitement</b>	Mise à jour de la liste des nouveaux utilisateurs Suppression des comptes non validés

# 7 - Sauvegardes, restaurations et maintenances

---

Ce chapitre décrit les opérations à exécuter pour sauvegarder et restaurer le système.

## 7.1 Sauvegarde du système

Toute la configuration du système est contenu dans la ZODB. Il s'agit du fichier Data.fs situé dans \$ZOPE\_HOME/var/

Par défaut, selon procédures standard hébergeur :

- Sauvegarde full serveur Netbackup le samedi soir puis incrémentale les autres soirs de la semaine.
- Dump des bases à chaud ou à froid selon la technologie, chaque nuit avec sauvegarde locale tournante sur X jours.

## 7.2 Restauration du système

Restauration Netbackup ou du Dump selon procédures standard hébergeur

## 7.3 Maintenance générale des composants et de l'application

Celle-ci s'effectue par la purge des fichiers journaux applicatifs. Tous les mois il est préconisé de faire une archive du dossier \$ZOPE\_HOME/log/ par la commande suivante et de l'archiver sur bande :

```
$ tar zcf $ZOPE_HOME/log/*.log vitamin2_log<AAAAMMJJ>.tgz
```

où le motif <AAAAMMJJ> prend le format suivant :

1. AAAA : année (exemple : 2005)
2. MM : mois (exemple : 05)
3. JJ : jour du mois (exemple : 05)

Supprimer les fichiers récemment archivés par la commande :

```
$ rm -fr $ZOPE_HOME/var/*.log
```

## 7.4 Page de Garde de la maintenance.

Pour des raisons de sécurité et de respect des utilisateurs, il est à signaler qu'une page de garde doit être mise en place pour toutes les applications du backoffice qui ne peuvent être accédées entre deux heures et 4 heures heure métropolitaine.

Celle-ci est disponible à l'url suivante :

<http://<adresseServeur>/html/AccueilMaintenance.html>

Ainsi les URL des url backoffice (<http://<adresseServeur>/Admin>) et de l'application XXX (<http://<adresseServeur>/XXX>) doivent être redirigés vers cette URL.

# 8 - Gestion des bases de données

---

Ce chapitre décrit les différentes opérations à mener pour gérer la base de données objet intégrée à Zope, cela comprend :

- Accès à la ZODB
- La gestion des données
- La sauvegarde de la base de données.

## 8.1 Accès à la ZODB

L'accès à la ZODB se fait en se connectant en administrateur à la ZMI à l'adresse `http://<ip_server>:<port>/manage` (Les identifiants de l'administrateur sont ceux saisis lors de la création de l'instance Zope)

Choisir ensuite le lien **Control Panel** puis **Database Management** puis **Main**

On accède à une interface permettant de gérer la ZODB

## 8.2 Gestion de données

A partir l'interface de gestion de la base de données, les actions suivantes :

- Faire un pack de la ZODB : efface les transactions précédentes sur les objets stockés (la commande UNDO ne permet plus de revenir à un état précédent)
- Visualiser l'activité récente (dernière heure) de la ZODB : nombre de connexions, objets chargés et enregistrés
- Accéder au paramètre du cache
- Effacer le cache

## 8.3 Sauvegarde de la base de données

<b>Sauvegarde</b>	
<b>Nom de la base</b>	ZODB
<b>Rôle</b>	
<b>Localisation</b>	\$ZOPE_HOME/var/Data.fs
<b>Taille initiale</b>	3 Go
<b>Taux d'accroissement</b>	Variable en fonction de l'activité en écriture et de la sauvegarde des événements sur les objets
<b>Procédure d'activation</b>	
<b>Fréquence de sauvegarde</b>	Quotidienne (peut être mis dans les batchs à heure fixe)
<b>Durée de rétention de sauvegarde</b>	1 semaine

## 8.4 Autres tâches de maintenance

Le script `cps_housekeeping.py` permet d'accomplir quelques tâches de maintenance sur une instance Zope active. Avant de lancer ce script, il faut prendre la précaution de faire un back-up du fichier `Data.fs`.

De manière classique, exécuter le script `cps_housekeeping.py` comme suit, avec les paramètres personnalisés, pour purger la ZODB des documents orphelins :

```
$ python /path/to/cps_housekeeping.py --purge-repository --
host=localhost --port=8080 --user=admin --password=admin --instance-
id=cps --zodbfile=/appli/zeo/var/Data.fs
```

De manière personnalisée, exécuter :

```
$ python /path/to/cps_housekeeping.py --host=localhost --port=8080 --
user=admin --password=admin --instance-id=cps
```

avec les options suivantes :

<code>-v</code> OR <code>--verbose</code>	: mode verbeux
<code>--days=NUMBER</code>	: indique le nombre de jours à garder dans l'historique pour un pack (0 par défaut)
<code>-r</code> OR <code>--purge-repository</code>	: efface les documents orphelins
<code>-l</code> OR <code>--purge-localroles</code>	: nettoie les rôles locaux en supprimant les membres effacés
<code>-p</code> OR <code>--pack-zodb</code>	: compacter la ZODB
<code>--zodbfile=CHEMIN</code>	: chemin complet vers la <code>Data.fs</code>
<code>-b</code> OR <code>--backup</code>	: faire un back-up de la ZODB (à utiliser en précisant
<code>--backupdir=CHEMIN</code>	: chemin utilisé pour stocker le backup de la ZODB

# 9 - Localisation physique des données et des programmes

---

Ce chapitre indique l'endroit où se trouvent réellement les données, les logs et les programmes.

Les programmes constituant la version de production sont situés sur le serveur dans le répertoire **\$ZOPE\_HOME** (par exemple dans /opt/vitamin2)

## 9.1 Liste des serveurs applicatifs installés

### 9.1.1 - Apache

- Répertoire de base d'installation : /etc/httpd/
- Répertoire de l'exécutable : /usr/sbin/apachectl (stop|start|restart)
- Répertoire de services : /etc/init.d/httpd (stop|start|restart)

### 9.1.2 - Zope

- Répertoire de base d'installation : \$ZOPE\_HOME
- Répertoire des exécutables :
- \$ZOPE\_HOME/bin/zopectl (start|stop|restart|status|debug)
- Répertoire de logs : \$ZOPE\_HOME/log/

## 9.2 Liste des composants installés pour l'application Vitamin2

Le répertoire des composants constituant Vitamin2 est : \$ZOPE\_HOME/Products

## 10 - Interfaces avec les autres systèmes

---

Ce chapitre décrit les opérations à faire pour que l'application puisse échanger des données avec d'autres systèmes.

L'application Vitamin2 utilise les listes de diffusion. Ce produit nécessite de copier sur le serveur de messagerie le script Python `stmp2zope.py` qui transmettra vers l'instance Zope les messages entrants des mailing lists. Le script devra être accessible pour l'utilisateur gérant la messagerie avec les droits d'exécution.

La configuration du serveur de messagerie est conservée avec les alias actuels permettant de rediriger les mails `@vitamin.fr` vers le serveur Zope. Une mise à jour sera peut-être nécessaire pour indiquer la nouvelle adresse du serveur dans l'alias de `vitamin2.fr`

[Peut-on apporter plus d'information par rapport à l'architecture finalement mise en place ?]

# 11 - Surveillance du système

---

Ce chapitre indique les opérations à effectuer pour surveiller et suivre le système ainsi que leur fréquence.

Pour une surveillance optimale de l'application en production, il est nécessaire de mettre en place trois types de surveillances :

- celle de l'infrastructure : essentiellement les serveurs (ping, CPU, mémoire, espace disque), les réseaux, les pare-feux ;
- celle des logiciels nécessaires à l'application : Apache, Zope ;
- celle applicative : analyse des journaux applicatifs.

## 11.1 Surveillance de l'infrastructure

Ce tableau récapitule les éléments d'infrastructure à surveiller sur chaque type de serveur :

<b>Composants</b>	<b>Élément à surveiller</b>
<b>Serveur Web</b>	- Indisponibilité
	- Charge CPU
	- Mémoire
	- Espace Disque
<b>Serveur d'applications et de données Zope</b>	- Disponibilité
	- Charge CPU
	- Mémoire
	- Espace Disque
<b>Pare feu</b>	- Indisponibilité
<b>Répartiteur de charge</b>	- Indisponibilité

## 11.2 Surveillance des logiciels

Ce tableau récapitule les éléments logiciels à surveiller sur les serveurs les hébergeant et une méthode de surveillance :

<b>Composants</b>	<b>Élément à surveiller</b>
<b>Apache</b>	Vérifier que le processus « Apache » est présent. Sur chaque serveur hébergeant un serveur Web Apache en activité, la commande suivante « # ps -eafww   grep apache » doit retourner quelque chose
<b>Zope</b>	Vérifier que le processus « Zope » est lancé. Sur le serveur, la commande suivante « # \$ZCOPE_HOME/bin/zopectl status » doit retourner le PID du processus en cours d'exécution

### 11.3 Surveillance applicative (fichiers journaux)

Il est nécessaire de mettre en place une analyse au fil de l'eau des fichiers journaux de l'application afin d'y détecter les dysfonctionnements qui se sont produits dans l'application.

La procédure à mettre en place est de détecter les lignes de ces fichiers contenant le mot ERROR ou Erreur ou FATAL afin qu'elles soient analysées par la production.

Le message associé à la ligne d'un message d'erreur fourni la cause du problème rencontré.

La liste des fichiers journaux à surveiller est la suivante :

<i>Nom du fichier</i>	<i>Description du contenu</i>
Sur les serveurs d'applications et de données : \$ZOPE_HOME/log/Z2.log	Traces de Zope (à analyser en cas de problèmes)
Sur le serveur de données où sont installés les batchs : /.../log/.../xxx.log	Traces des batchs (à analyser en cas de problème)

### 11.4 Les URL de surveillance

Il existe plusieurs serveurs mis en jeu pour faire fonctionner l'application Vitamin2. En voici les différentes URL permettant de contrôler le bon fonctionnement de chacun :

#### **URL de test pour le serveur Web Apache :**

http://<ip\_serveur>/html/ok.html => Ceci doit afficher le mot OK si tout va bien.

#### **URL de test pour le serveur Zope :**

http://<ip\_serveur>:<port>/ZopeOk => Ceci doit afficher le mot OK si tout va bien, sinon rien.

#### **URL de test pour tester l'applicatif :**

http://<ip\_serveur>:<port>/<objet\_CPS>/VitaminOk => Ceci doit afficher le mot OK si tout va bien, sinon rien.