ZCP 7.1 (build 40326)

Plateforme Collaborative Zarafa

Manuel de l'administrateur Zarafa

ZCP 7.1 (build 40326) Plateforme Collaborative Zarafa
Manuel de l'administrateur Zarafa
Édition 7.0

La Plateforme Collaborative Zarafa (ZCP) allie l'ergonomie d'Outlook à la stabilité et la flexibilité d'un serveur Linux. Elle comporte une interface Web exhaustive, le WebAccess de Zarafa, et offre des options ingénieuses d'intégration avec une multitude de clients, y compris toutes les plateformes mobiles les plus populaires.

La majorité des composants de ZCP sont Open Source, sous licence *AGPLv3*[1], et peuvent donc librement être téléchargés avec *la version communautaire ZCP*[2]. xTESTx

Plusieurs composants propriétaires existent, principalement :

[4] http://creativecommons.org/licenses/by-sa/3.0/
[1] http://www.gnu.fr/licenses/agpl-3.0.html
[2] http://http://www.zarafa.fr/content/community

- le client Windows de Zarafa, permettant l'intégration Outlook,
- l'intégration BES de Zarafa, permettant la connectivité BES (Blackberry Enterprise Server),
- le plug-in ADS de Zarafa, permettant l'intégration Active Directory, et
- les utilitaires de sauvegarde de Zarafa.

Ces composants, ainsi que plusieurs fonctionnalités avancées destinées aux larges environnements et aux hébergeurs, sont uniquement disponibles avec le contrat de support attaché aux *éditions commerciales de ZCP*[3].

Par ailleurs, une large sélection d'offres ZCP hébergé est disponible.

Ce document, le manuel de l'administrateur, décrit l'installation, la mise à jour, la configuration et la maintenance de ZCP sur votre serveur Linux. Par ailleurs, diverses options d'intégration et de configuration avancée sont également traitées.

[3] http://www.zarafa.fr/contenu/éditions

Introduction

La plate-forme collaborative Zarafa (ZCP) est une suite logicielle Open Source permettant de remplacer Microsoft Exchange. Son architecture très modulaire s'appuie autant que possible sur les standards classiques et s'intègre parfaitement avec les composants Open Source communément répandus.

Ce document décrit les tâches courantes liées à l'administration de ZCP.

Important

Bien que nous nous efforcions à Zarafa de maintenir les informations contenues dans ce manuel aussi correctes que possible, nous nous réservons le droit de les modifier à tout moment sans aucun avertissement préalable.

1.1. Audience ciblée

Ce manuel est destiné aux administrateurs système responsables de l'installation, de la maintenance et du déploiement de ZCP. Nous présumons que les lecteurs de ce manuel bénéficient d'une compréhension approfondie dans les domaines suivants :

- Concepts et tâches de l'administration système Linux
- Normes du courrier électronique
- Concepts de sécurité
- Services d'annuaire
- Gestion de bases de données

1.2. Architecture

En accord avec la philosophie UNIX, ZCP est constituée de plusieurs composants qui gèrent chacun une tâche bien précise. Voir *Figure 1.1, « Diagramme de l'architecture de la Plateforme Collaborative Zarafa (ZCP) »* pour une description des interactions ente les composants et les protocoles utilisés. Ce diagramme décrit une configuration classique du type utilisé par la plupart de nos clients. Seuls les composants les plus répandus sont utilisés dans ce diagramme.

La partie supérieure du diagramme présente les clients : les applications logicielles avec lesquelles les utilisateurs accèdent à leurs données. Certains de ces logiciels sont des applications d'ordinateurs, tandis que d'autres sont des applications mobiles.

Entre “l'Internet” et le “Serveur Zarafa”, se trouvent les composants de l'infrastructure de Zarafa (en bleu) ainsi d'autres composants d'infrastructure communément répandus et non spécifiques à Zarafa (en gris). Ces composants permettent de faciliter la communication entre le Serveur Zarafa et les différents clients. Microsoft Outlook n'a besoin d'aucune infrastructure particulière, mais communique directement avec le Serveur Zarafa à l'aide du client Windows de Zarafa.

Globalement, le Serveur Zarafa émets des appels MAPI, et stocke ses données dans une base MySQL. Plusieurs méthodes sont disponibles pour l'authentification des utilisateurs (et sont détaillées dans ce document), les plus communes étant l'utilisation de serveurs qui implémentent LDAP (p. ex.: OpenLDAP, ou Microsoft Active Directory).

La section suivante décrit succinctement chacun des composants de ZCP.

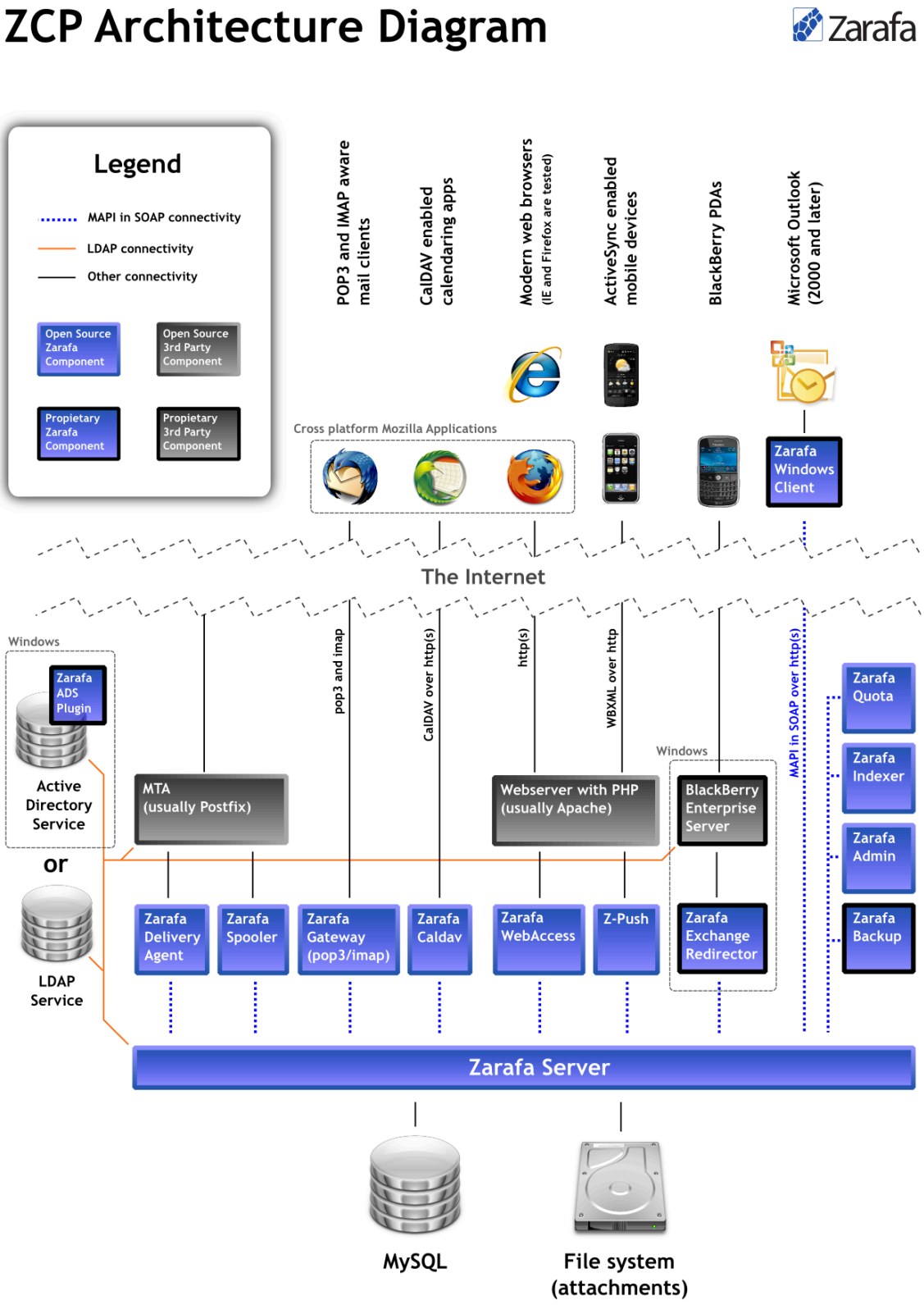


Figure 1.1. Diagramme de l'architecture de la Plateforme Collaborative Zarafa (ZCP)

1.3. Composants

Une installation de la Plateforme Collaborative Zarafa comporte généralement les composants suivants :

- **Le Serveur Zarafa** (**`zarafa-server`**) — Le serveur accepte les connexions de tous les clients par protocole SOAP (HTTP), et stocke ses données dans une base SQL.
- **Le gestionnaire de Licence Zarafa** (**`zarafa-licensed`**) — Le gestionnaire de licence vérifie les fonctionnalités qui seront disponibles selon que la suscription choisie soit celle de l'édition PME, Professional ou Enterprise.
- **Le client Windows de Zarafa** — Le client Zarafa fournit un accès à Microsoft Outlook à l'aide d'une interface MAPI. Les connexions avec le serveur sont gérées par SOAP.
- **Le WebAccess de Zarafa** (**`zarafa-webaccess`**) — Une interface Web riche en fonctionnalités (avec un aspect et un ressenti semblable à Microsoft Outlook) qui permets à ses utilisateurs de travailler en collaboration à partir de n'importe quel ordinateur disposant d'une connexion Internet.
- **Zarafa WebApp** (**`zarafa-webapp`**) — La nouvelle génération de client Web collaboratif, qui intègre des outils de messagerie instantanée et de visioconférence.
- **L'agent de distribution Zarafa** et **le gestionnaire de file d'attente Zarafa** (**`zarafa-dagent`**, **`zarafa-spooler`**) — Les utilitaires qui distribuent la messagerie électronique au monde extérieur. Le dagent (contraction de 'delivery agent' pour agent de distribution) distribue le courriel reçu par l'agent de transport de courrier (MTA) à l'utilisateur Zarafa. Le gestionnaire de file d'attente envoie le courriel en attente dans la queue d'envoi au MTA spécifié.
- **L'utilitaire d'administration Zarafa** (**`zarafa-admin`**) — L'utilitaire d'administration en ligne de commande, permet la gestion des utilisateurs, de leurs données et de leurs groupes.
- **La passerelle Zarafa** (**`zarafa-gateway`**) — Service optionnel fournissant POP3 et IMAP aux utilisateurs Zarafa.
- **Le contrôleur Zarafa** (**`zarafa-monitor`**) — Le service qui contrôle les répertoires des utilisateurs pour excès de quota.
- **Le gestionnaire Caldav de Zarafa** (**`zarafa-caldav`**) — Service optionnel permettant la prise en charge iCal et CalDAV. CalDAV est recommandé pour sa rapidité et ses transferts de données moins importants.
- **L'utilitaire de sauvegarde Zarafa** (**`zarafa-backup`**, **`zarafa-restore`**) — Un utilitaire de sauvegarde par bloc individuel permettant de générer des sauvegardes de base de stockage en toute simplicité et de restaurer (une partie de) ces sauvegardes ultérieurement. Cette partie est uniquement disponible dans les versions commerciales de Zarafa.
- **Zarafa search** — Service optionnel fournissant une indexation en texte intégral. Ceci permet d'effectuer des recherches rapides parmi les courriels et les pièces jointes.
- **Apache** — Fournit les pages Web de WebAccess au navigateur de l'utilisateur.
- **PHP** — WebAccess est écrit dans ce langage de programmation.
- **L'extension PHP-MAPI** — Module d'extension pour PHP, permettant l'utilisation de la couche MAPI. Grâce à cette extension, les fonctionnalités MAPI deviennent accessibles aux développeurs PHP. En pratique, cela signifie que des clients Web MAPI peuvent être conçus. Ce qui est le cas pour WebAccess.

- **L'extension Python-MAPI** — Module d'extension pour Python, permettant l'utilisation de la couche MAPI.. Grâce à cette extension, les fonctionnalités MAPI deviennent accessibles aux développeurs Python.

Pour la connectivité avec les périphériques mobiles nous recommandons l'utilisation de *Z-Push*[1] (see *Section 5.5, « Configure Z-Push (Remote ActiveSync for Mobile Devices) »*), une implémentation Open Source du protocole ActiveSync. Pour les périphériques mobiles plus anciens, ou pour les périphériques mobiles que n'utilisent pas le protocole ActiveSync, nous fournissons l'utilitaire **Zarafa WebAccess Mobile** (**`zarafa-webaccess-mobile`**) qui offre une interface Web de base et des fonctionnalités limitées. Veuillez noter que ce composant n'est pas officiellement approuvé et qu'il sera probablement retiré des versions futures de ZCP.

1.4. Protocoles et Connexions

Toutes les applications qui se connectent au Serveur Zarafa utilisent MAPI encapsulé dans SOAP pour cela (voir le diagramme de l'architecture de Zarafa). Même WebAccess utilise MAPI encapsulé dans SOAP (fournit par l'extension PHP-MAPI) pour se connecter au Serveur Zarafa.

Le client Zarafa est un fournisseur MAPI classique, compatible avec Microsoft Windows. Il se connecte au serveur (MAPI encapsulé dans SOAP) par protocole HTTP(S).

1.4.1. SOAP

SOAP est l'abréviation de Simple Object Access Protocol. C'est un protocole d'échange de données qui exécute des procédures d'appel à distance (RPC) entre diverses applications au travers d'un réseau, notamment sur Internet.

SOAP est basé sur XML et HTTP 1.1 (port **`80`**, ou port **`443`** dans le cas de HTTPS). Grâce à ces standards, il est possible de se connecter de manière transparente par serveur proxy, à la plupart des réseaux sans aucune modification.

1.4.2. HTTP sécurisé (HTTPS)

Le client Windows de Zarafa peut créer une connexion HTTP sécurisée au serveur avec SSL (HTTPS). Lorsqu'un profil MAPI de Outlook est créé, il est possible de paramétrer la connexion afin qu'elle utilise HTTPS. Toutes les connexions passant par le réseau seront alors chiffrées, rendant toute indiscrétion pratiquement impossible.

Le Serveur Zarafa doit être paramétré afin qu'il puisse également accepter les connexions SSL. Par défaut, cette option n'est pas activée, car elle nécessite la création de certificats SSL. Une fois le certificat du serveur créé, les connexions SSL pourront être directement acceptées à partir d'un client. En outre, les autres composants de Zarafa (comme l'agent de distribution Zarafa ou le gestionnaire de file d'attente de Zarafa) peuvent également établir une connexion HTTPS au serveur et s'authentifier à l'aide de la clé privée du Serveur Zarafa.

[1] http://z-push.sourceforge.net

1.5. Éditions ZCP et licences

1.5.1. La souscription d'évaluation

Si la version d'évaluation est utilisée, un délai est offert durant lequel les fonctionnalités complètes de ZCP pourront être testées. Il sera ensuite possible de continuer à utiliser la base de données qui a été générée par cette version d'évaluation après l'installation d'une souscription commerciale valide.

Une version d'évaluation peut être souscrite sur *http://www.zarafa.com/serial_request*.

1.5.2. L'édition Communautaire de The ZCP

L'édition Communautaire de la Plateforme Collaborative Zarafa est distribuée sous licence *Affero GPLv3*[2]. Cette édition peut être utilisée au maximum par 3 utilisateurs avec le client Windows propriétaire de Zarafa (afin de se connecter avec Microsoft Outlook). WebAccess, la passerelle IMAP et la synchronisation peuvent être utilisés par un nombre illimité d'usagers.

Note

Le composant de gestion de licence doit être en service pour pouvoir utiliser la prise en charge de Microsoft Outlook support dans l'édition Communautaire. Une souscription n'est cependant pas nécessaire.

1.5.3. Les éditions commerciales de ZCP

L'usage des éditions PME, Professional, Enterprise ou Hébergé nécessitent une souscription commerciale. Il sera précisé dans ce document lorsqu'une fonctionnalité ou un composant n'est pas disponible sans édition commerciale.

1.5.4. Utilisateurs actifs et non-actifs

Les souscriptions ZCP s'accordent pour une base nominale d'utilisateur. Une souscription de base est prévue pour un nombre fixe d'utilisateurs, qui peut être étendu par l'addition de Licences supplémentaires d'accès Client (CAL) ; p. ex. une souscription de base pour 10 utilisateurs, accompagnée d'une CAL pour 10 utilisateurs, est fonctionnellement équivalente à une souscription de base pour 20 utilisateurs.

Les souscriptions sont basées sur des utilisateurs nommés ; autrement dit, 10 utilisateurs nommés peuvent être ajoutés dans un système possédant une licence pour 10 utilisateurs. Cependant, il y a également des utilisateurs qui n'apparaissent pas directement dans ce décompte ; ce sont les utilisateurs soit-disant ‘non-actifs’. Un exemple d'utilisateur non-actif est l'utilisateur ‘info’ ou ‘support’ . Cet utilisateur est considéré comme tel dans le sens où il est capable de recevoir des courriers et a accès à tous les dossiers standards, mais il ne lui sera pas permis de s'authentifier. Ce sont d'autres utilisateurs qui ouvriront la base de stockage 'info' en tant que délégués et qui pourront récupérer les courriels qu'elle contient.

Chaque souscription donne droit à un certain nombre d'utilisateurs non-actifs supplémentaires. Le nombre d'utilisateurs non-actifs autorisés est égal à 150% du nombre des utilisateurs d'une souscription, avec un minimum de 20 utilisateurs non-actifs. Le nombre d'utilisateurs non-actifs autorisés a été augmenté à partir des versions 6.40.8 et 7.0.0 afin de permettre la création de base de

[2] http://www.zarafa.com/content/affero-gplv3

stockage d'archive non-actives (Précédemment, le maximum autorisé des utilisateurs non-actif était limité à 50 %).

Exemples :

- Souscription : 10 utilisateurs
- Utilisateurs actifs : 10
- Utilisateurs non-actifs : 20
- Souscription : 400 utilisateurs
- Utilisateurs actifs : 400
- Utilisateurs non-actifs : 600

Si tous les utilisateurs actifs ne sont pas utilisés, il est possible d'utiliser leur surplus comme compte non-active.

Note

Un utilisateur est défini comme ‘actif’ ou ‘non-actif’ au moment de sa création. Il est uniquement possible de basculer le type d'un utilisateur actif en non-actif ou vice-versa depuis la version 6.40 de ZCP : dans les versions précédentes, l'utilisateur devait être supprimé puis recréé avec un type différent.

Dans les configurations LDAP, l'indicateur non-actif des utilisateurs peut être contrôlé à l'aide de la directive de configuration **`ldap_nonactive_attribute`**. Avec l'utilisation du composant DB, l'indicateur **`non-active`** se spécifie à l'aide de l'option **`-n`** de l'utilitaire **`zarafa-admin`** lors de la création d'un utilisateur. Le composant 'Unix user' se sert du shell Unix de l'utilisateur tel qu'il est défini dans **`/etc/passwd`** afin de déterminer si la base de stockage doit être non-active.

Installation

2.1. Configuration requise

2.1.1. Configuration matérielle recommandée

To give an estimate on the resource use of ZCP we have created the table below. These are merely guidelines, giving a rough estimation on what hardware is required. In this table we assume the CPU is under low load from other applications and size concerns the storage used in MySQL Server for the mailboxes.

Tableau 2.1. Configuration matérielle recommandée

Size of all mailboxes/ Users	CPU (Cores)*	Mémoire	Disque dur	Niveau Raid
< 5 GB / 1-25 users	2	2 Go	SATA, SAS	RAID1, 7.2K
> 5 - < 10 GB / 26-50 users	4	4 Go	SAS	RAID1, 7.2K
> 10 - < 20 GB / 51-100 users	4	6 GB	SAS	RAID10, 7.2K
> 20 - < 50 GB / 101-200 users	6	8 GB	SAS	RAID10, 10K
> 50 GB - < 100GB / 201-300 users	6	10 GB	SAS	RAID10, 10K
> 100GB - < 250 GB / 301-500 users	6	12 GB	SAS	RAID10, 10K
> 250 GB / 501-1000 users	8	16 Go	SAS	RAID10, 15K or SSD/7.2K Hybrid

Important

Tuning of server configuration and the software components on the specific onsite usage can drastically improve performance of your ZCP instance. For more then 500 users and larger total mailbox storage size then 250Gb and/or any high availability structures the recomendations are highly influenced and its advised to seek professional engineering support.

2.1.2. Plateformes prises en charge

ZCP est constitué d'un large éventail de composants : certains composants qui forment son architecture et qui sont exécutés sur des plateformes Linux et d'autres composants qui peuvent être installés sur les ordinateurs des utilisateurs finaux. Dans cette section, nous détaillerons les différentes plateformes supportées.

Au début de chaque cycle de diffusion générale (comme 6.x.x or 7.x.x) nous décidons des plates-formes devant être prises en charge. Généralement ce sont les versions actuelles d'une plate-forme

ainsi que celle qui l'a immédiatement précédé. Pendant le cycle d'une version majeure, de nouvelles plates-formes peuvent être prises en charges, mais aucune ne peut être enlevée.

Veuillez utiliser les paquets logiciels x86_64 ou 64bit si vous disposez d'un matériel et d'un système d'exploitation 64bit. Il est recommandé d'utiliser la version 64bit si possible.

Important

Le support pour l'architecture **`ia64`** sera abandonné dans le cycle ZCP-7.x.x

Tableau 2.2. Plateformes supportées pour les composants de l'architecture ZCP

Version OS	Architectures CPU prises en charge
RHEL 5	i386, x86_64, ia64*
RHEL 6	i686, x86_64
SLES 10	i586, x86_64, ia64*
SLES 11	i586, x86_64, ia64*
Debian 5.0 (Lenny)	i386, x86_64, ia64*
Debian 6.0 (Squeeze)	i386, x86_64
Ubuntu 8.04 LTS (Hardy)	i386, x86_64
Ubuntu 10.04 LTS (Lucid)	i386, x86_64
Ubuntu 12.04 LTS (Precise)	i386, x86_64

Tableau 2.3. Plateforme supportées pour les applications **Windows Client**, **Migration Tool** et **ADS Plugin** de ZCP.

Édition MS Windows	Architectures CPU prises en charge
Windows Server 2003	32bit, 64bit
Windows Server 2008	32bit, 64bit
Windows XP	32bit, 64bit
Windows Vista	32bit, 64bit
Windows 7	32bit, 64bit

Ce sont les plateformes Microsoft Windows prises en charge par les composants qui nécessitent une plateforme Microsoft Windows, c'est-à-dire le client Windows, l'utilitaire de migration et le plugin ADS.

Note

L'utilitaire de migration n'est pour l'instant pas disponible sur la plateforme 64bit.

Pour plus d'information au sujet des navigateurs officiellement pris en charge, des clients Outlook ou des niveau de support technique, veuillez consulter *le document sur le cycle de vie du support des produits*[1].

[1] http://doc.zarafa.com/trunk/Support_Lifecycle_Policy/en-US/html-single

2.1.3. Dépendances

Afin de pouvoir constituer ou installer les composants de l'architecture ZCP, un nombre de conditions doivent être remplies. Voici les dépendances principales de ZCP :

- **MySQL**, sans l'accès à un serveur MySQL le serveur Zarafa ne pourra pas fonctionner. Il n'est pas obligatoire que le serveur MySQL soit exécuté depuis la même machine que le serveur Zarafa Server, ce n'est donc pas une dépendance de paquet logiciel. La version 4.0 ou inférieure de MySQL ne fonctionnera pas correctement. ZCP a été testé avec MySQL 4.1, 5.0 et 5.1.
- **Apache** ou tout autre serveur Web qui prend en charge le PHP. ZCP a été testé avec Apache 2.0 et 2.2.
- **PHP**, seul comme CGI ou, de préférence comme module de serveur Web. ZCP a été testé avec PHP 4.3.x et les dernières versions 5.x.
- **Libicu** bibliothèque fournissant un support Unicode complet et robuste.
- **SMTP** serveur au choix. ZCP a été testé avec Postfix, Sendmail et Qmail
- **LDAP** serveur au choix (optionnel pour l'administration des utilisateurs). Zarafa a été testé avec OpenLDAP, eDirectory et Microsoft Active Directory.
- **Catdoc** utilisé pour l'indexation de texte provenant de documents Office.
- **Poppler-utils** utilisé pour l'indexation de texte provenant de documents pdf.
- **w3m** utilisé pour l'indexation de texte HTML provenant du courriel.

La majorité de ces dépendances sont résolues automatiquement par le gestionnaire de paquets logiciels de la distribution Linux sur laquelle ZCP a été installé. Ceci permet aux composants utilisés par ZCP d'être installés et mis à jour automatiquement à l'aide du gestionnaire de paquet logiciel de la distribution. Certaines dépendances du tableau ci-dessus sont des dépendances d'exécution, elles devront donc être installées manuellement car elles ne doivent pas nécessairement être exécutées sur la même machine.

La méthode habituelle pour déployer ZCP consiste à installer les paquets logiciels sur l'une des distributions Linux prise en charge, permettant ainsi au composant tiers utilisés par ZCP d'être installés automatiquement par le gestionnaire de paquets logiciel de la distribution. Dans ce cas de figure, les composant tiers seront mis à jours selon la méthode habituelle employée par la distribution.

Note

Si vous utilisez Debian ou Ubuntu, et que vous démarrez avec une nouvelle installation *fraiche*du serveur, vous pouvez utiliser **tasksel** afin d'installer un ensemble LAMP (Apache, MySQL, PHP) complet. Cet ensemble apportera tous les paquets logiciels requis pour que le script d'installation de Zarafa puisse s'exécuter avec succès.

2.2. Installation

Il y a 4 façons principales d'installer ZCP: (1) à l'aide du gestionnaire de paquets logiciels de la distribution, (2) à l'aide de notre script d'installation, (3) en installant manuellement chaque paquet logiciel, et (4) à partir du code source. Dans cette section, chacune de ces méthodes sera expliquée avec ses avantages et ses inconvénients.

Note

Dans l'édition communautaire, le paquet **`zarafa-licensed`** n'est pas indispensable, sauf pour y activer la prise en charge d'Outlook auquel cas, il est nécessaire de lancer le daemon **`zarafa-licensed`**.

Note

L'agenda multi-utilisateur contenu dans le paquet **`zarafa-webaccess-muc`** est une fonctionnalité qui n'est pas disponible dans la version Communautaire Une souscription valide est nécessaire.

Note

Les bibliothèques partagées qui fournissent le composant utilisateur, sont installées dans**`/usr/lib64/zarafa`**, au lieu de l'emplacement **`/usr/lib/zarafa`**. Ce chemin doit être ajusté dans le fichier de configuration **`server.cfg`**. Définir **`plugin_path`** sur **`/usr/lib64/zarafa`**, afin que le serveur puisse accéder aux fichiers du plugin d'administration des utilisateurs

Note

The MySQL option **`max_allowed_packet`** should not be set higher than 128M. This can conflict with Zarafa offline mode in Outlook. If the MySQL option must be higher you must also update the Zarafa offline clients. Change the value **`max_allowed_packet`** in **`C:\Program Files (x86)\Zarafa\Zarafa Outlook Client\MySQL\My.ini`** on the client.

2.2.1. Installation à l'aide du script d'installation

Lorsque ZCP est téléchargé à partir du site Web *http://www.zarafa.com/* website (l'édition communautaire ou bien une des éditions commerciales) un fichier tarball se présente avec le contenu suivant :

- les paquets (RPMs ou DEBs selon la distribution)
- les scripts **`install.sh`** et **`uninstall.sh`** (et un fichier additionnel **`helpers.inc`**)
- un dossier nommé **`windows`** contenant les binaires se rapportant à Windows
- un dossier nommé **`browsers`** contenant le plugin Glisser&Déposer de Firefox

Le script **`install.sh`** effectuera automatiquement les actions décrites dans la section *Installation manuelle* ci-dessous. Ainsi il effectuera les actions suivantes :

- vérification des dépendances logicielles
- installation des paquets
- vérification de l'accès à la base de données MySQL

- demande des options de configuration

Le script d'installation est invoqué à l'aide de la commande :

```
sh ./install.sh
```

Après l'exécution de **`install.sh`**, le serveur devrait être prêt à démarrer. Il faut alors créer des bases de stockage comme cela sera indiqué par le script.

Si le script **`install.sh`** est invoqué avec le paramètre **`-config`**, alors il n'installera aucun paquet, mais ne fera que demander les options de configuration.

```
sh ./install.sh -config
```

Le script **`install.sh`** configure toujours le serveur afin d'utiliser le plugin d'administration des utilisateurs DB. Si une autre système d'administration utilisateur doit être utilisé, veuillez consulter *Chapitre 4, Configure ZCP Components* pour les informations de configuration du serveur.

Note

Si une version plus ancienne de ZCP est installée, veuillez consulter *Chapitre 3, Mise à jour*. Le script **`install.sh`** n'est **pas** utilisable dans ce cas.

2.2.2. Installation manuelle des paquets

Veuillez utiliser les paquets adaptés à la distribution utilisée. Consulter la liste des distributions dans *Section 2.1.2, « Plateformes prises en charge »*. Pour les distributions non listées, il est possible d'utiliser les paquets de la distribution s'en rapprochant le plus, cependant, Zarafa ne peut pas se porter garant de telles installations.

Les paquets disponibles sont affichés dans le tableau suivant :

Tableau 2.4. Paquets logiciels disponibles

Nom du paquet	Description
libical	Contains the ical library used for Caldav and iCal
libvmime	Contains the library for working with mime and rfc822 messages
libkyotocabinet16	Contains the library of routines for managing the full text search database
php-mapi	Contient l'extension php-mapi
python-mapi	Contient les liaisons Python MAPI pour Zarafa
python-zcp-license	Contient les liaisons Python pour la gestion de la licence Zarafa
zarafa	Peut être utilisé afin d'installer un système ZCP complet sur le serveur

Nom du paquet	Description
zarafa-backup	Contient les utilitaires de sauvegarde et de restauration de Zarafa
zarafa-client	Contient le fournisseur MAPI pour les clients MAPI
zarafa-dagent	Contient l'agent de distribution
zarafa-gateway	Contient la passerelle POP3/ IMAP
zarafa-ical	Contient la passerelle iCAL/ Caldav
zarafa-libarchiver	Contient la bibliothèque de reconstruction des 'stubs' pour Zarafa Archiver
zarafa-libs	Contient les bibliothèques de conversion pour courriel et agenda
zarafa-licensed	Contient les binaires non Open Source et leurs fichiers de configuration
zarafa-search	Contient le moteur de recherche en texte intégral
zarafa-monitor	Contient le contrôleur de quota
zarafa-multiserver	Contient les bibliothèques multi-serveur
zarafa-search	Contains the full text search component
zarafa-server	Contient le serveur principal et ses fichiers de configuration
zarafa-spooler	Contient le spooler
zarafa-utils	Contient les utilitaires d'administration, tels que zarafa-admin et zarafa-fsck
zarafa-backup	Contient l'utilitaire de sauvegarde par boîte individuelle
zarafa-webaccess	Contient WebAccess
zarafa-webaccess-muc	Contient l'agenda multi-utilisateur de WebAccess
zarafa-webapp	Contient WepApp, la nouvelle interface Web en remplacement de WebAccess
zarafa-archiver-extra	Contains additional licensed archiver tools

Note

Veuillez ne pas mélanger de paquets logiciels provenant de distributions différentes ! Veuillez ne choisir qu'une distribution et n'utiliser uniquement que ses paquets logiciels. Si cette règle n'est pas respectée, des erreurs s'ensuivront !

2.2.2.1. Distributions de type RPM

Utiliser les commandes suivantes afin d'installer les paquets ZCP sur les distributions de type RPM :

```
rpm -Uvh <package file>
```

Remplacer **<package file>** avec les paquets contenus dans le tarball. Commencer par **libvmime**, **libical** et **zarafa** (dans cet ordre) puis installer les autres paquets. Le gestionnaire de paquet pourrait trouver des dépendances non résolues, dans ce cas, tenter d'installer les paquets pour ces dépendances de la manière habituelle pour cette distribution (**yum -i** on Red Hat, **zypper -i** on OpenSUSE/SLES).

2.2.2.2. Distributions de type DEB

Pour les distributions de type DEB (généralement Debian et Ubuntu) exécuter :

```
dpkg -i <package file>
```

Pour installer correctement les dépendances pour ZCP **apt-get** ou tout autre utilitaire équivalent peut être utilisé.

Pour MySQL, exécuter :

```
apt-get install winbind
```

Pour Apache et le support PHP nécessaire, exécuter :

```
apt-get install apache2-mpm-prefork libapache2-mod-php5
```

Si les packages Zarafa ne s'installent pas à cause des dépendances, alors il faudra utiliser la commande suivante afin de les installer :

```
apt-get -f install
```

Si Apache est installé avec la prise en charge de PHP après que les paquets de Zarafa aient déjà été installés, il faudra utiliser la commande suivante afin d'actualiser automatiquement la configuration de PHP :

```
dpkg-reconfigure zarafa
```

2.2.2.3. Installer à partir du code source

ZCP n'est pas officiellement supporté par Zarafa lorsqu'il a été compilé à partir du code source, cependant dans certaines situations - c'est-à-dire en utilisant ZCP sur des environnements non pris en charge, ou lors de la préparation de correctifs pour ZCP - il peut être très utile de procéder à une

installation à partir code source. Comme la plupart des composants de ZCP sont distribués sous licence Open Source (AGPLv3), chacun a le droit de compiler ZCP à partir du code source.

Cependant, la procédure exacte pour compiler ZCP à partir du code source dépasse la portée de ce document. Cette procédure est sensiblement différente pour chaque distribution et est sujette à changement. Veuillez consulter notre *wiki*[2] (effectuer une recherche avec les mots clés 'from source') pour les dernières informations à ce sujet.

2.3. Résolution de problèmes d'installation

2.3.1. Processus du serveur

Make sure at least MySQL 5.0 is installed. The server will only run with this version of the database server or a more recent version.

S'il y a des problèmes lors du chargement des bibliothèques ou si la connexion à MySQL échoue, des messages d'erreurs seront envoyés au fichier de journalisation. Veuillez toujours vous assurer que le service soit correctement démarré.

Si une option de configuration incorrecte est présente dans un fichier de configuration, le service ne démarrera pas. Les options inexactes seront affichées sur le terminal.

2.3.2. WebAccess & WebApp

Pour que WebAccess puisse s'afficher correctement, les extensions PHP suivantes sont nécessaires :

- **`gettext`**
- **`session`**
- **`iconv`**
- **`xml`**

Certaines distributions fournissent par défaut la prise en charge de ces extensions dans leur paquet logiciel PHP. Pour la distribution SUSE, ces modules sont fournis dans des RPMs séparés, par exemple :

```
php5-gettext-5.2.8-37.4.x86_64.rpm
php5-iconv-5.2.8-37.4.x86_64.rpm
```

Ces versions peuvent différer dans les éditions plus récentes de SUSE.

Pour les distributions Red Hat Enterprise Linux et les distributions de type Debian, ces modules sont fournis dans le package PHP standard qui a déjà été installé, à cause des dépendances.

Si vous rencontrez des problèmes avec les envois de pièces jointes, veuillez vous assurer que le serveur Web est autorisé à créer des fichiers dans le dossier **`WebAccess/tmp`**. Si un utilisateur est immédiatement déconnecté dès qu'il tente de s'identifier sur WebAccess, veuillez vous assurer que PHP est configuré avec :

```
register_globals = off
```

[2] http://wiki.zarafa.com/

Si une distribution est utilisée en combinaison avec SELinux, un message d'erreur peut apparaitre lors de la connexion à WebAccess. Le message par défaut indique que le mot de passe saisi est erroné ou que le serveur Zarafa ne fonctionne pas. Quand SELinux est activé, il bloque la connexion du WebAccess vers le serveur Zarafa. La politique de contrôle d'accès SELinux Zarafa permettant cette connexion est disponible ici : *http://www.zarafa.com/wiki/index.php/Zarafa_Selinux_policy*.

ou bien il est possible de désactiver SELinux à l'aide de la commande suivante :

```
setenforce permissive
```

Si SELinux doit être complètement désactivé, alors il sera également nécessaire de modifier le fichier **`/etc/sysconfig/selinux`** afin qu'il soit également désactivé après un redémarrage.

Plus d'informations à propos de SELinux sont disponibles sur *http://fedora.redhat.com/docs/selinux-faq*.

Mise à jour

3.1. Préparation

Avant de mettre à jour ZCP vers une nouvelle version, il est recommandé d'effectuer une sauvegarde de la base de données et des fichiers de configuration.

Note

Lors de la mise à jour d'une édition commerciale de ZCP vers une nouvelle version majeure, par exemple de 6.40.x vers 7.0.x, la clé de souscription doit être convertie. La conversion d'une clé de souscription s'effectue sur *notre portail*[1].

Premièrement, arrêter l'exécution du service MTA sur votre serveur. Ainsi aucun courriel ne sera perdu si une erreur devait se produire au cours de la mise à jour. Dans le cas de Postfix, utiliser la commande :

```
/etc/init.d/postfix stop
```

Ensuite arrêter les services activés afin que la base de données ne soit plus en fonctionnement :

```
/etc/init.d/zarafa-spooler stop
/etc/init.d/zarafa-server stop
/etc/init.d/zarafa-licensed stop
```

De même avec les services optionnels, s'ils avaient également été lancés :

```
/etc/init.d/zarafa-dagent stop
/etc/init.d/zarafa-gateway stop
/etc/init.d/zarafa-ical stop
/etc/init.d/zarafa-indexer stop
/etc/init.d/zarafa-search stop
/etc/init.d/zarafa-monitor stop
```

Important

Lorsque les pièces jointes sont stockées dans la base de données, une mise à jour vers une version 6.30.x ou ultérieure augmentera le fichier de stockage de la base de données par la taille combinée de toutes les pièces jointes (étant stockées dans la table “lob”). Au cours de la mise à jour, une table temporaire est créée afin de stocker toutes les pièces jointes avant d'être supprimée, et puisqu'il n'est pas possible de réduire le fichier de stockage de la base de donnée, la taille de ce fichier augmentera de la taille combinée de toutes les pièces jointes qui y sont stockées.

Des informations supplémentaires sur la migration des pièces jointes depuis la base de données vers le système de fichier sont disponibles sur *notre wiki*[2].

[1] https://portal.zarafa.com/
[2] http://www.zarafa.com/wiki/index.php/Store_attachment_outside_of_the_database

3.2. Création de sauvegardes

Maintenant il faut créer des sauvegardes de la base de données et des fichiers de configuration. Faire une copie du répertoire **`/etc/zarafa`**, qui contient tous les fichiers de configuration.

```
cp -r /etc/zarafa /etc/zarafa.bck
```

As Zarafa stores attachments of items on the filesystem, make a copy of the attachment directory.

```
cp -r /var/lib/zarafa /var/lib/zarafa.bck
```

Pour sauvegarder les base de données MySQL, un **`mysqldump`** peut être exécuté :

```
mysqldump --single-transaction -p zarafa > zarafa.sql
```

ou bien le répertoire complet des données MySQL peut être copié :

```
/etc/init.d/mysqld stop
cp -r /var/lib/mysql /var/lib/mysql.bck
cp -r /etc/my.cnf /etc/my.cnf.bck
```

Note

The paths could be different when default configuration is changed.

3.3. Dépendances ZCP7

Une fois les sauvegardes effectuées avec succès, le pack logiciel de Zarafa peut être mis à jour. Quelques nouvelles dépendances devront être résolues avant de pouvoir effectuer cette mise à jour.

Tableau 3.1. Dépendances ZCP7

Distribution	Dépendances
Debian 5	libboost-filesystem1.35.0, libboost-system1.35.0, libicu38, w3m, python-mysqldb
Debian 6	libboost-filesystem1.42.0, libboost-system1.42.0, libicu44, w3m, python-mysqldb
RHEL5	libicu, w3m, MySQL-python
RHEL6	boost-filesystem, boost-system, libicu, w3m, MySQL-python
SLES10	libicu, w3m, python-mysql
SLES11	libicu, w3m, python-mysql
Ubuntu 8.04	libicu38, w3m, python-mysqldb
Ubuntu 10.04	libboost-filesystem1.40.0, libboost-system1.40.0, libicu42, w3m, python-mysqldb

3.4. Effectuer la mise à jour pour une distribution de type RPM

Une fois les sauvegardes créées, la mise à jour peut s'effectuer comme n'importe quelle installation manuelle. Pour une installation de type RPM, exécuter la commande suivante :

```
rpm -Uvh <package name>.rpm
```

Note

Il n'est pas nécessaire d'installer **`zarafa-licensed`** pour l'édition Communautaire. L'exécution du daemon zarafa-licensed n'est nécessaire que si l'intégration de Microsoft Outlook est utilisée.

Une fois les nouveaux packs logiciels installés, les fichiers d'exemple de configuration files qui se trouvent dans le répertoire **`/usr/share/doc/zarafa/example-config`** peuvent être consultés pour les nouvelles options de configuration. Les derniers changements sont également consultables dans les *Notes de version*[3].

3.5. Effectuer la mise à jour pour une distribution de type Debian

Décompresser le fichier tarball : tar zxvf zcp-7.0.0rc1-26667-debian-6.0-i386-free.tar.gz

Installer la nouvelle version de libvmime 0.9 qui est incluse avec Zarafa :

```
dpkg -Bi libvmime0_0.9.2*
```

Installer la version de libical qui est incluse avec Zarafa :

```
dpkg -Bi libical0_0.44*
```

Installer la version de python-mapi qui est incluse avec Zarafa :

```
dpkg -i python-mapi*
```

Pour les installations de type Debian, exécuter la commande suivante pour la mise à jour de ZCP :

```
dpkg -Bi <nom_du_pack_logiciel>
```

Selon la version exacte du pack logiciel 6.x qui est installée, cette commande peut retourner des erreurs pour les packs “zarafa” et “zarafa-licensed”. En raison du morcellement poussé et du changement des noms de packs logiciels, certains conflits ne peuvent pas être directement résolus à l'aide de “dpkg”. Si vous recevez un message d'erreur au cours de la mise à jour de ces packs, il faudra tenter une nouvelle installation à l'aide de la commande suivante :

```
dpkg -i <nom_du_pack_logiciel>
```

Ou bien exécuter cette commande :

[3] http://doc.zarafa.com/trunk/Release_Notes/en-US/html/_config_file_changes.html

```
apt-get install -f
```

ce qui devrait résoudre la situation.

Lorsque vous serez invité à faire des choix à propos des fichiers de configuration Zarafa modifiés, la meilleure option dépendra en grande partie de votre situation particulière.

Note

Il n'est pas nécessaire d'installer **`zarafa-licensed`** pour l'édition Communautaire. L'exécution du daemon zarafa-licensed n'est nécessaire que si l'intégration de Microsoft Outlook est utilisée.

Une fois les nouveaux packs logiciels installés, les fichiers d'exemple de configuration files qui se trouvent dans le répertoire **`/usr/share/doc/zarafa/example-config`** peuvent être consultés pour les nouvelles options de configuration. Les derniers changements sont également consultables dans les *Notes de version*[4].

3.5.1. Étapes de mise à jour pré 6.40

Il y a eu certaines modifications de configuration depuis les versions 6.40 et ultérieures afin de prendre en charge les nouvelles fonctionnalités du carnet d'adresse global, comme les contacts, les groupes dynamiques et les groupes de sécurité. Surtout avec l'utilisation du composant LDAP, le serveur ne redémarrera pas correctement sans que des modifications aient été apportées au fichier de configuration LDAP. Si le composant DB ou le composant Unix est utilisé, aucune modification des fichiers de configuration n'est nécessaire. Cependant, il peut être utile de les regarder afin de pouvoir configurer les nouvelles options.

Please check the upgrade page on *our wiki*[5] for up-to-date upgrade details.

Pour prendre en charge correctement les contacts provenant de Microsoft Active Directory, la valeur du champ de configuration **`ldap_user_unique_attribute`** doit être changée de **`objectSid`** en **`objectGuid`**. Comme cette valeur est attachée à l'identifiant unique des utilisateurs, sa modification sans une mise à jour de la base de données entrainera une suppression de tous les utilisateurs existant par le serveur Zarafa qui recréera ensuite ce qu'il aura détecté comme des nouveaux utilisateurs. Ceci doit absolument être évité, c'est pourquoi il est indispensable d'exécuter le script **`db-upgrade-objectsid-to-objectguid.pl`** qui se trouve dans le dossier **`/usr/share/zarafa/doc/`**. Ce script détectera les paramètres LDAP définis dans le fichier existant **`/etc/zarafa/server.cfg`** et affectera la nouvelle identification unique à la base de données. Après l'exécution du script, il est également nécessaire de modifier le fichier de configuration LDAP afin que le nouvel attribut d'identification unique soit utilisé. S'assurer que le processus du serveur Zarafa ne soit pas en cours d'exécution lors du lancement de ce script.

Note

Si OpenLDAP est utilisé, il n'est pas nécessaire de modifier l'attribut **`ldap_user_unique_attribute`**.

[4] http://doc.zarafa.com/trunk/Release_Notes/en-US/html/_config_file_changes.html
[5] http://www.zarafa.com/wiki/index.php/Upgrading_to_6.40

Les paramètres LDAP 'Envoyer en tant que' sont incompatibles avec celles des versions 6.30 depuis la version 7.0. Cette modification a été effectuée afin d'étendre la permission envoyer-en-tant-que aux groupes Si les options envoyer-en-tant-que sont activées pour les utilisateurs, le script **`ldap-switch-sendas.pl`** devra être exécuté. Ce script mettra à jour le serveur LDAP ou ADS avec les permissions envoyer-en-tant-que actuelles et les basculera au format 6.40.

```
cd /usr/share/doc/zarafa
chmod 755 ldap-switch-sendas.pl
./ldap-switch-sendas.pl
```

Dans 6.40, les permissions envoyer-en-tant-que sont définies sur l'utilisateur. Par exemple : Un utilisateur non-actif **`info@company`** existe et certains utilisateurs doivent utiliser cette adresse dans l'entête 'De :' pour envoyer du courrier. Ces utilisateurs sont ajoutés à l'objet **`info@company`** dans la liste d'attribut envoyer-en-tant-que.

Dans la configuration LDAP, les options de base de recherche séparée pour chaque objet sont combinées dans un filtre de recherche nommé **`ldap_search_base`**. Toutes les anciennes options **`search_base`** devront être supprimées. De même, toutes les options d'étendue de recherche être supprimées.

Ensuite, les types d'objet devront être définis. Ceci s'effectue généralement à l'aide de l'attribut **`objectClass`**. Chaque objet utilisateur doit être défini à l'aide de son **`objectClass`**.

Finalement, l'ancien filtre de recherche par objet sera vidé comme c'est en double. Il est cependant toujours conseillé d'utiliser **`zarafaAccount`** dans le filtre utilisateur, cette option est donc toujours disponible.

Pour prévenir la suppression des utilisateurs par le serveur, une option en mode protégé est disponible dans le fichier de configuration **`server.cfg`**. L'activation de cette option bloquera toutes les créations et les suppressions d'utilisateurs et de groupes.

Ajouter l'option suivante dans le fichier **`/etc/zarafa/server.cfg`** afin d'activer le mode protégé :

```
user_safe_mode = yes
```

Vérifier le fichier de journalisation du serveur après le démarrage du serveur Zarafa Server pour détecter les modifications affectant les utilisateurs. Si aucun utilisateur n'a été recréé ou supprimé, le fichier de configuration a correctement effectué sa tâche et l'option **`user_safe_mode`** peut désormais être désactivée en toute sécurité.

Important

Il est fortement recommandé de n'utiliser le mode protégé qu'à la suite d'une mise à jour. Une fois la mise à jour effectuée avec succès, le mode protégé devrait être désactivé. L'activation du mode protégé dans un système en production peut causer des problèmes de performance.

Note

Au cours de la mise à jour de ZCP 6.30 vers ZCP 7.0 il n'est pas nécessaire de passer par une mise à jour préalable vers la version 6.40.

3.5.2. De la version 6.40 vers la version 7.0.0 ou ultérieure

En raison de la masse des données devant être converties lors d'une mise à jour vers ZCP 7.0 et du délai probablement important nécessaire à sa mise en œuvre, le serveur refusera par défaut de mettre à jour la base de données.

La mise à jour de la base de données de Zarafa nécessite plusieurs heures au minimum, veuillez noter que le système Zarafa ne peut pas être utilisé au cours de ce processus. Pour fournir une estimation de la durée de cette mise à jour, nous avons créé le script upgrade-calculation qui doit être lancé sur le serveur en version 6.40. Cette estimation est approximative et nous nous efforçons de l'affiner régulièrement à l'aide des retours d'expérience de la communauté. Télécharger le script ici : *http://www.zarafa.com/upgrade*

Au cours de votre mise à jour, la mesure du temps réellement écoulé par rapport aux valeurs estimées nous sera d'une grande utilité. Merci de bien vouloir nous faire connaitre le type des données que vous avez mises à jour afin de nous permettre d'optimiser ce script.

Important

Veuillez vous assurer que les paramètres innodb de votre serveur MySQL sont correctement optimisé. Pour plus d'information à propos des paramètres principaux d'optimisation des performances MySQL, veuillez consulter *Chapitre 9, Réglage des performances*.

Pour mettre à jour la base de données, il est recommandé d'employer l'utilitaire**`zarafa7-upgrade`** inclus dans le paquet logiciel zarafa-server de ZCP 7.0. Cet utilitaire de mise à jour effectuera toutes les étapes de mise à jour nécessaires et vous tiendra informé du progrès de la procédure. L'utilitaire **`zarafa7-upgrade`** se trouve dans le dossier /usr/share/doc/zarafa et requiert les dépendances **`python-mysqldb`** ou **`MySQL-python`**, ainsi que **`python-mapi`** packages. Ce dernier peut être trouvé dans le tarball ZCP.

Avant que le script **`zarafa7-upgrade`** ne soit lancé, le service **`zarafa-server`** doit être démarré afin de convertir la base de données vers la dernière révision de celle-ci pour la version 6.40.

```
/etc/init.d/zarafa-server start
```

Consulter le fichier de journalisation **`/var/log/zarafa/server.log`** pour la progression de la mise à jour.

```
[root@zarafa ~]# tail -f /var/log/zarafa/server.log
Mo 27 Feb 2012 09:50:48 CET: Starting zarafa-server version 7,0,5,31880, pid 30725
Mo 27 Feb 2012 09:50:48 CET: Connection to database 'zarafa' succeeded
Mo 27 Feb 2012 09:50:48 CET: WARNING: zarafa-licensed not running, commercial features will
 not be available until it's started.
Mo 27 Feb 2012 09:50:48 CET: Start: Move IMAP subscribed list from store to inbox
Mo 27 Feb 2012 09:50:55 CET: Done: Move IMAP subscribed list from store to inbox
Mo 27 Feb 2012 09:50:55 CET: Start: Update sync table time index
Mo 27 Feb 2012 09:50:58 CET: Done: Update sync table time index
Mo 27 Feb 2012 09:50:58 CET: Start: Update changes table state key
Mo 27 Feb 2012 11:05:12 CET: Done: Update changes table state key
Mo 27 Feb 2012 11:05:12 CET: Start: Converting database to Unicode
Mo 27 Feb 2012 11:05:12 CET: Will not upgrade your database from 6.40.x to 7.0.
Mo 27 Feb 2012 11:05:12 CET: The recommended upgrade procedure is to use the zarafa7-upgrade
 commandline tool.
Mo 27 Feb 2012 11:05:12 CET: Please consult the Zarafa administrator manual on how to
 correctly upgrade your database.
Mo 27 Feb 2012 11:05:12 CET: Alternatively you may try to upgrade using --force-database-
upgrade,
```

```
Mo 27 Feb 2012 11:05:12 CET: but no progress and estimates within the updates will be
 available.
Mo 27 Feb 2012 11:05:12 CET: Failed: Rollback database
Mo 27 Feb 2012 11:05:12 CET: Can't update the database: Unable to upgrade zarafa from version
 6.40.30778 to 7.0.5.31880
Mo 27 Feb 2012 11:05:12 CET: Server shutdown complete.
```

Une fois la base de données convertie vers la structure correcte, le serveur Zarafa s'arrêtera automatiquement et un message s'affichera pour prévenir que la mise à jour doit être continuée manuellement à l'aide du script zarafa7-upgrade. Exécuter le script **`zarafa7-upgrade`** afin de convertir la disposition de la base de données et la préparer au traitement Unicode.

Sur Debian et Ubuntu le fichier doit au préalable être décompressé :

```
gunzip /usr/share/doc/zarafa/zarafa7-upgrade.gz
python /usr/share/doc/zarafa/zarafa7-upgrade
```

Pour lancer l'utilitaire de mise à jour, exécuter :

```
[root@zarafa ~]# python /usr/share/doc/zarafa/zarafa7-upgrade
Converting search folders to Unicode: 879 / 879 (100%)
Converting properties for IO performance: 69318024 / 69318024 (100%)
Creating counters for IO performance: 16 / 16 (100%)
Creating common properties for IO performance: 4 / 4 (100%)
Creating message attachment properties for IO performance: 2 / 2 (100%)
Creating tproperties for IO performance: 69318023 / 69318023 (100%)
Converting hierarchy for IO performance: 69318023 / 69318023 (100%)
Creating deferred table for IO performance: 1 / 1 (100%)
Converting changes for IO performance: 56266424 / 56266424 (100%)
Converting names table to Unicode: 10331 / 10331 (100%)
```

Le script convertira les tables de la base de données en UTF-8 afin d'être intégralement compatible avec l'Unicode et convertira la base de données vers la nouvelle disposition utilisée par ZCP 7.0. Le script affichera les progrès effectués pas la mise à jour comme indiqué ci-dessus.

Une méthode alternative consiste à forcer le serveur à mettre à jour la base de données en le démarrant avec l'option --force-database-upgrade.

Important

L'utilisation de l'option --force-database-upgrade n'est pas recommandée car elle n'affiche aucune indication des progrès réalisés et ne peut être interrompue.

Note

Lors d'une mise à jour depuis les anciennes versions de ZCP, par exemple ZCP 6.30.x, zarafa-server mettra premièrement à jour la base de données vers la disposition ZCP 6.40 à la suite de quoi le script de mise à jour pourra être exécuté.

3.5.3. De la version 7.0 vers la version 7.1.0 ou ultérieure

The **`zarafa-indexer`** has been replaced by the **`zarafa-search`** package. Make sure you remove **`zarafa-indexer`** when upgrading to 7.1 and install the **`zarafa-search`** package. You can remove the old index directories and files as they won't be used anymore. All directories found in

the index_path location (default: /var/lib/zarafa/index/) can be removed. The new **zarafa-search** application only creates .kct files and will not interfere with the old index files.

The **zarafa-search** options in the server.cfg file have also changed. All the old indexer options are replaced by new search options. The following config options can be removed from the old server config file:

```
index_services_enabled
index_services_path
index_services_search_timeout
```

These options are replaced by the following search options:

```
search_enabled = yes
search_socket = file:///var/run/zarafa-search
search_timeout = 10
```

These options are by default set, so there is no need to change these config values to use the new zarafa-search engine after the upgrade.

When using Debian or Ubuntu, please check if the file **/etc/default/zarafa** contains the following lines at the end.

```
# set to no to disable zarafa-search at startup
SEARCH_ENABLED=yes

# Location of the configuration files
SEARCH_CONFIG=/etc/zarafa/search.cfg

# Additional options that are passed to the Daemon.
SEARCH_OPTS=""
```

If these lines are not available, the **zarafa-search** service will not start automatically. The lines can be manually added or the file can be overwritten by the file provided in the package.

```
mv /etc/default/zarafa.dpkg-dist /etc/default/zarafa
```

ZCP 7.1 introduces stored procedures in MySQL to improve streaming speed used in the **zarafa-search** and for offline users. This changes the privileges zarafa-server needs to correctly use the MySQL database. The mysql user needs the CREATE PROCEDURE privilege, which can be given using the GRANT sql command. Please see *Chapitre 4, Configure ZCP Components* for a full list of all required privileges and grant examples.

Besides this the "enable_sql_procedures" option must be enabled in the server.cfg

The SQL Procedures allow for some optimized queries when streaming with enhanced ICS. This is default disabled because you must set *thread_stack = 256k* in your MySQL server config under the [mysqld] tag and restart your MySQL server.

Note

Note that any search indexes made with prior releases of 7.1.0 (RC or beta) need to be dropped before use with the final or RC3.

3.6. Finalisation de la mise à jour

Après vérification des nouvelles options de configuration, les services pourront de nouveau être redémarrés :

```
/etc/init.d/zarafa-server start
/etc/init.d/zarafa-spooler start
/etc/init.d/zarafa-licensed start
```

Les services optionnels pourront également être redémarrés :

```
/etc/init.d/zarafa-dagent start
/etc/init.d/zarafa-gateway start
/etc/init.d/zarafa-ical start
/etc/init.d/zarafa-search start
/etc/init.d/zarafa-monitor start
```

Comme les mise à jour incluent généralement une extension différente de **`php-mapi`**, le serveur Web devra également être redémarré :

```
/etc/init.d/apache2 restart
```

ou

```
/etc/init.d/httpd restart
```

ZCP 7.0 comporte une nouvelle passerelle IMAP/POP3 améliorée. La nouvelle passerelle offre une meilleure compatibilité et une performance accrue grâce à l'utilisation d'informations supplémentaires stockées dans la base de données et dans le répertoire des pièces jointes. Comme ces informations supplémentaires utilisent davantage d'espace disque et ne sont utilisées que lorsque les utilisateurs se connectent par IMAP, les fonctionnalités IMAP/POP3 sont **`désactivées`** par défaut.

Si des utilisateurs devaient accéder à l'IMAP ou au POP3, ces fonctionnalités devraient être manuellement activées. Pour plus d'information sur l'activation/désactivation de fonctionnalités, voir *Section 8.7, « Gestion des fonctionnalités Zarafa »*.

Le script **`optimize-imap.py`** permettra de générer une version IMAP optimisée de tous les messages existants. En exécutant ce script pour tous les courriers existant, la structure de l'enveloppe et la structure du corps seront stockés dans la base de données. De plus, un fichier complet du message RFC822 message seront générés et stockés en compression gzip dans le répertoire des pièces jointes.

Le script générera ces données uniquement pour les utilisateurs pour lesquels les protocoles IMAP ou POP3 ont été activés.

Pour lancer ce script, veuillez exécuter la commande suivante :

```
/usr/share/zarafa-gateway/optimize-imap.py
```

Note

En ce qui concerne les nouveaux courriels reçus sur ZCP 7.0, les données IMAP optimisée sont automatiquement stockées dès lors que les utilisateurs ont les protocoles IMAP ou POP3 activés.

Configure ZCP Components

Most ZCP and 3rd party components are configured by a configuration file. This section explains most common options that are set to get these components up and running. It is important to note that components usually have to be restarted to make use of updated configuration files, read more about this in the *Chapitre 7, Gestion des services ZCP*.

In short, after modifications have been made to a component's configuration file, that component has to be restarted with:

```
/etc/init.d/zarafa-<component name> restart
```

4.1. Configure the Zarafa Server

The Zarafa Server component is configured by a system-wide configuration file, usually located here:

```
/etc/zarafa/<component name>.cfg
```

When installing ZCP an example of this file is put here:

```
/usr/share/doc/zarafa-<component name>/example-config/zarafa-<component name>.cfg
```

The options and their default values are explained both by the in-line comments of the example file and in the following manual page:

```
man <component name>.cfg
```

For example:

```
man zarafa-server.cfg
```

If a line is not present, the default setting will be assumed. For most basic setups the defaults of the example file will work fine. In this chapter we only explain the basic configuration option of Zarafa Server.

The Zarafa Server needs a MySQL database to function, and therefor needs to know how to connect to the MySQL server and the authentication credentials for its database. It will create a database and the tables it needs at first start.

Make sure that the MySQL user that the Zarafa Server uses to connect to the database has all privileges, including the right to create a new database. Also make sure to give the user enough permissions to connect from localhost to this database, or --if the Zarafa server connects over the network to the MySQL database-- allow it to connect from the IP-address from which the Zarafa Server will connect.

For example the following MySQL statement grants all privileges to user “zarafa” with password “password” from localhost:

```
GRANT ALL PRIVILEGES ON zarafa.* TO
'zarafa'@'localhost' IDENTIFIED BY 'password';
```

If you want to restrict the privileges of the zarafa connection, the following grant command lists only the required privileges:

```
GRANT alter, create, create routine, delete, drop, index, insert, lock tables, select, update
 ON zarafa.* TO
'zarafa'@'localhost' IDENTIFIED BY 'password';
```

To configure the Zarafa Server to use the MySQL server the options starting with **`mysql`** in the **`zarafa-server.cfg`** need to be set. Once this is setup the Zarafa Zerver should start normally.

4.2. Configure language on RPM based distributions

After the creation of new users the Zarafa Server will automatically create the actual mailbox. This mailbox is by default created in the language of the Linux server. When another language is required the following configuration file has to be changed:

```
/etc/sysconfig/zarafa
```

Change the option **`ZARAFA_USERSCRIPT_LOCALE`** to the correct language, for example **`nl_NL.UTF-8`** or **`fr_FR.UTF-8`**.

In order to use this language setting make sure the language packs are installed. Red Hat and SuSE based systems contain all language packs by default.

The option **`ZARAFA_LOCALE`** in the **`/etc/sysconfig/zarafa`** file can be used to start the Zarafa Server component in the correct language. This language setting is used to set the default options, like the Public Folder name to the correct language.

The WebAccess GUI language can be set at the login screen. This can be configured per user login.

Important

When upgrading from an earlier ZCP version, please review the language settings as from ZCP 7.0.0 the locale has to be set in UTF-8.

4.3. Configure language on Debian based distributions

When adding new users the Zarafa Server will automatically create the actual mailbox. The mailbox is by default created in english language. To create the mailboxes in english it's required to have the en_US.UTF-8 locale installed.

When the mailbox should be created in another language the following configuration file has to be changed:

```
/etc/default/zarafa
```

Change the option **`ZARAFA_USERSCRIPT_LOCALE`** to the correct language, for example **`nl_NL.UTF-8`** or **`fr_FR.UTF-8`**.

In order to use this language setting make sure the language packs are installed or configured.

To install a language pack Ubuntu based systems, use the following command (this example is for the Dutch -nl pack):

```
apt-get install language-pack-nl
```

On Debian based systems the locale settings need to be enabled from the locale list. Use the following command to enable the different locales:

```
dpkg-reconfigure locales
```

The option **ZARAFA_LOCALE** in the **/etc/default/zarafa** file can be used to start the Zarafa Server component in the correct language. This language setting is used to set the default options, like the Public Folder name to the correct language.

The WebAccess GUI language can be set at the login screen. This can be configured per user login. For non-English WebAccess languages the appropriate language-packs need to be installed as well.

Important

When upgrading from an earlier ZCP version, please review the language settings as from ZCP 7.0.0 the locale has to be set in UTF-8.

At Debian distributions the entry in /etc/apache2/envvars needs to be set to force the locale, else locale specific characters might not be displayed correctly in the webaccess.

```
## The locale used by some modules like mod_dav
#  export LANG=C
## Uncomment the following line to use the system default locale instead:
. /etc/default/locale
```

4.4. User Authentication

Another important configuration option for the Zarafa Server is the **user_plugin**. This setting determines which back-end is used for managing users and groups. There are four options, namely **db**, **unix** and **ldap** and **ldapms**.

By default the **db** plugin is used as it does not require any further configuration. The **ldap** plugin is used most in larger setups as it proves to be most flexible and integrates nicely with an organization's the existing infrastructure.

The **ldapms** plugin is required when configuring a multi-server Zarafa environment. Multi-server support is only available in the Enterprise edition.

More information on managing users can be found in *Chapitre 8, Gestion des utilisateurs*.

For a comparison between the different plugins, see the table below:

Tableau 4.1. An example table

Feature	DB	Unix	LDAP	LDAPMS
Create/delete/ modify users	X	X	X	X
Set aliases	On MTA level	On MTA level	X	X
Hide users			X	X
Sendas permissions	X	X	X	X
Multi-server support				**X**

Feature	DB	Unix	LDAP	LDAPMS
Sendas permissions of groups			X	X
Security Groups	X	X	X	X
Distribution groups			X	X
Hide groups			X	X
Dynamic groups			X	X
Contacts support			X	X
Multi-tenancy support	X		X	X
Addresslists support			X	X
Multi-server support				**X**

4.4.1. The DB Authentication Plugin

This plugin uses the Zarafa MySQL database to store user and group information. The **`zarafa-admin`** tool can be used to manage users.

The DB plugin supports only basic user and group information. For more advanced configurations, we advise to use the LDAP plugin.

For more information about user management with the **`zarafa-admin`** tool, see *Chapitre 8, Gestion des utilisateurs*.

4.4.2. The Unix Authentication Plugin

The Unix plugin is used on a server which has all its user information setup in the **`/etc/passwd`** file. Group information will be read from **`/etc/group`**. Passwords are checked against **`/etc/shadow`**, so the **`zarafa-server`** process must have read access to this file (this process is normally run as root, so usually that is not a problem).

Since the unix files do not contain enough information for Zarafa, there are some properties of a user that will be stored in the database. These properties are the email address, overriding quota settings, and administrator settings. The **`zarafa-admin`** tool has to be used to update these user properties. All other user properties are done using the normal unix tools.

A configuration file, **`/etc/zarafa/unix.cfg`**, exists for this plugin. The default set by this file are usually enough, in-line comments explain each option. In this configuration file the **`uid`** range of users wanted in the Zarafa server needs to be defined. The same goes for the groups.

Non-active users are appointed by a specific shell, default **`/bin/false`**. These users cannot login, but the stores can be opened by other users. An administrator should setup the correct access rights for these stores.

For an overview of all configuration options of the unix authentication plugin, use:

```
man zarafa-unix.cfg
```

4.4.3. The LDAP Authentication Plugin

The LDAP plugin is used for coupling any LDAP compliant server with the Zarafa Server. This way, all users, groups and membership information can be retrieved ‘live’ from an LDAP server.

The LDAP plugin support next to the default users, groups and companies also the following object types:

- **Contacts** — External SMTP contacts which can be used as members of distribution lists
- **Addresslists** — Sub categories of the Global Address Book, based on a specified LDAP filter
- **Dynamic groups** — Dynamically created groups, based on a specified LDAP filter. Therefore LDAP plugin is the recommended user plugin for ZCP.

The Zarafa Server needs two configuration directives in the **`server.cfg`** configuration file to use the LDAP backend, namely:

```
user_plugin = ldap
user_plugin_config = /etc/zarafa/ldap.cfg
```

The defaults for OpenLDAP and for Active Directory can be found in the **`/usr/share/doc/zarafa/example-config`** directory. Based on these examples the **`/etc/zarafa/ldap.cfg`** file should be adjusted to configure the LDAP authentication plugin.

More details about configuring the LDAP plugin with OpenLDAP, see *Section 5.2, « Configuration de l'intégration ZCP OpenLDAP »* or *Section 5.3, « Configuration de l'intégration ZCP Active Directory »* for Active Directory.

4.5. Autoresponder

ZCP contains an autoresponder that can be used when a user is out of the office to reply automatically to all incoming e-mails. The autoresponder will automatically be spawned whenever an e-mail is delivered by **`zarafa-dagent`** to a store that has the ‘Out of Office’ option turned ON.

Users can manage the autoresponder of their own store as well as of stores to which one has at least secretary rights. Note that this includes public folders. Please refer to the User manual on how to manage these settings.

To prevent autoresponder loops (e.g. when sending automated responses to an automated response, which in turn sends an automated response, etc), the autoresponder will only send one autoresponse message per day for any unique sender e-mail address. The autoresponder will also not respond in any of the following cases:

- Sending an out-of-office message to yourself.
- Original message was to *mailer-daemon*, *postmaster* or *root*.
- Original message was from *mailer-daemon*, *postmaster* or *root*.

Furthermore, the autoresponder is configured by default to respond only to e-mails in which the user was explicitly mentioned in the ‘To’ header. This means that e-mails that were received because the user was in the ‘Cc’ header or because the user was in a distribution group, are not responded to.

Most behaviour can be configured by editing the file **`/etc/zarafa/autorespond`**. This file contains the following settings, which will be used for all autorespond messages server-wide:

```
AUTORESPOND_CC=0
```

Set this value to ‘1’ to allow autoresponding to messages in which the recipient was only stated in the ‘Cc’ header.

```
AUTORESPOND_NORECIP=0
```

Set this value to ‘1’ to autorespond to all messages, even if the recipient is not stated in any header (for example when the email was directed at a mailing list or group)

```
TIMELIMIT=$[24*60*60]
```

Sets the minimum number of seconds between autoresponses to the same e-mail address

The following settings normally do not need to be modified:

```
SENDDB=${TMP:-/tmp}/zarafa-vacation-$USER.db
```

(file which stores the last date of sending per email address)

```
SENDDBTMP=${TMP:-/tmp}/zarafa-vacation-$USER-$$.tmp
```

(temporary file used during update of the database)

```
SENDMAILCMD=/usr/sbin/sendmail
```

(command used to send actual vacation message)

```
SENDMAILPARAMS="-t -f"
```

(parameters used to send actual vacation message)

If an alternate autoresponder is required, please refer to the **`zarafa-dagent`** manual page which describes how to use an alternate script (using the **`-a`** option).

4.6. Storing attachments outside the database

Since ZCP version 6.0 it is possible to save the attachments outside the database. ZCP 7.0.5 and higher will use the filesystem as default location for attachment storage.

For first time installations, the attachment storage method should be selected before starting the server for the first time as it is not easy to switch the attachment storage method later on.

To change the attachment storage location, edit the following option in the **`/etc/zarafa/server.cfg`**.

```
attachment_storage = files
attachment_path = /var/lib/zarafa/attachments
```

For upgrades, a script exists that copies the attachments from the database to the file storage. This script can be found in **`/usr/share/doc/zarafa`**, and is named **`db-convert-attachments-to-files`**. This script can be used as follows:

```
db-convert-attachments-to-files <mysqluser> <mysqlpass> <mysqldb> <destination path> [delete]
```

Note

The script can be executed while the zarafa-server process is running.

It is only possible to convert from database storage to file storage. The **<delete>** switch is optional. If this parameter is given, the attachments are also removed from the database. Keep in mind that during the conversion the storage of the attachments on the harddisk will double. The amount of storage in MySQL used by ZCP can be looked up the with the following MySQL statements:

```
mysql> use zarafa;
mysql> show table status;
```

Check the **data_length** column for the lob table. This contains the number of bytes needed for the attachment storage.

To select this new storage method, change the **attachment_storage** option in the **server.cfg** file and point the **attachment_path** option to the folder where the attachments should be stored. After changing this option **zarafa-server** needs to be started once with the **--ignore-attachment-storage-conflict** parameter.

Advantages of attachments outside the database are:

- MySQL does not save the large binary blobs in the database. This improves the general read and write access.
- Attachments will not cause cache purges of MySQL.

Disadvantages of attachments outside the database are:

- A MySQLdump of the database is not enough for a full recovery.
- Remote storage of attachments requires a new system, like folder mounted through NFS or Samba.

Important

It is very important, when choosing to store the attachments outside the database, to update the backup strategy accordingly.

4.7. SSL connections and certificates

The Zarafa Server is capable of directly accepting encrypted SSL connections.

This feature may already be available when the HTTPS Apache server is setup to proxy these connections to the Zarafa Server.

However, having native SSL connections to the server has an interesting advantage: Zarafa components running beyond localhost can login using their SSL certificate.

This section will describe how to setup certificates to add native SSL connections to Zarafa.

First, we will create the directory to contain the certificate and setup the permissions, since it contains our private key.

```
mkdir /etc/zarafa/ssl
chmod 700 /etc/zarafa/ssl
```

If Zarafa is run as another user, as described in the Running as non-root user section, do not forget to chown the directory as well.

Now we are ready to create a *Certificate Authority* (CA). This CA will be used to create the server certificate and sign it. We provide a **ssl-certificates.sh** script in the **/usr/share/doc/zarafa** directory, which uses the **openssl** command and the **CA.pl** script from OpenSSL. Depending on the distribution used this script can be installed in different directories. The script will try to find it on its own. If it is not found, either OpenSSL is not installed, or the script is in an unknown location, and location of the script has to be provided manually. Normally, the **ssl-certificates.sh** script can be run without problems.

```
cd /etc/zarafa/ssl
sh /usr/share/doc/zarafa/ssl-certificates.sh server
```

The parameter server is added, so the name of the new certificate will be called **server.pem**. When the CA is not found in the default **./demoCA** directory, it needs to be created. By pressing enter, the creation of the new CA is started.

Enter a password (passphrase) when asked for. This is the password used later on to sign certificate requests. Then certificate information should be entered. Do not leave the **Common Name** field blank, otherwise the creation will fail.

Now that we have a CA, we can create *self-signed* certificates. The **ssl-certificates.sh** script will automatically continue with this step. Enter a password for the request, and enter the certificate details. Some details need to be different from those typed when the CA was created. At least the field **Organizational Unit Name** needs to be different. The challenge password at the end may be left empty.

This step created a Certificate Request, that needs to be signed by the CA that was created in the first step of the script. Type the password of the CA again when asked for. The details of the certificate will be shown, and asked for acceptance. Accept the certificate.

As the last step, the public key of this certificate will be offered. Since the server certificate just was created the public key of this certificate is not needed.

Now that the the CA certificate and the server certificate have been created, SSL can be enabled in the **server.cfg** file, which is normally disabled. The port **237** is set for SSL connections. This port number can be changed if necessary.

```
server_ssl_enabled = yes
server_ssl_port = 237
```

The CA certificate must be set in the **server_ssl_ca_file** setting. The server certificate and password must be set in the **server_ssl_cert_file** and **server_ssl_cert_pass** options.

```
server_ssl_ca_file = /etc/zarafa/ssl/demoCA/cacert.pem
server_ssl_key_file = /etc/zarafa/ssl/server.pem
server_ssl_key_pass = <password>
```

Restart the **zarafa-server** process, and now it's possible to connect directly to the SSL port. Create a new Outlook profile, and mark the SSL connection option. Set the port to **237**. The connection to the server has now been encrypted.

4.8. Configure the License Manager

Note

With the ZCP opensource edition the License Manager is not needed.

The License Manager (**zarafa-licensed**) expects **/etc/zarafa/license** to contain a file named **base** which simply holds the license key. To install a subscription key, use the following command:

```
mkdir -p /etc/zarafa/license
echo <subscription key> > /etc/zarafa/license/base
```

<subscription key> should be replaced with a valid subscription key obtained from Zarafa or one of its partners.

Note

The subscription key consists only of numbers and capital letters.

If an extra CAL (Client Access License) is also available, the key can be added with:

```
echo 'CAL key' > /etc/zarafa/license/cal1
```

If more than one CAL are available, please install one CAL per file in the license directory. The filename of the CAL is of no importance. Sub-folders in the **/etc/zarafa/license** folder are not allowed.

4.9. Configure the Zarafa Spooler

The Zarafa-spooler sends email from the global outgoing queue to a SMTP server, which sends the email to the correct address.

When an email message is sent from Outlook or WebAccess, the message is placed in the Outbox folder, and a submit message is sent to the Zarafa server. The server notifies the Zarafa spooler to send the email to the SMTP server. The spooler will now start to convert the message to a normal email message. When the conversion is complete, a connection to the supplied SMTP server is created, and the email is sent to the SMTP server.

The spooler will send the email, and after the mail is sent, will move the mail automatically to the user's Sent Items folder.

If at any time an error was found, the user will be notified with an 'Undeliverable' message. The message will contain an error description on which error was found. Often, the user can retry to send the message.

Note

Both external and internal emails will be sent via the MTA.

4.9.1. Configuration

The Spooler is configured the same as the server. Options in the spooler configuration file are the name or ip-address of the SMTP server, where to find the Zarafa server, and logging options.

```
smtp_server
```

The name or IP-address of the SMTP server, which will send the email to the destination. This server may also be given as an argument when starting the spooler.

```
server_socket
```

The UNIX socket of the Zarafa server. The spooler will use this socket to create a connection to the server. This value should be the same as set in the server configuration file. The default value is **`/var/run/zarafa`**.

```
[logging]
```

The spooler has the same configuration options as the server to configure logging options.

For an overview of all the configuration options of the **`zarafa-spooler`**, use:

```
man zarafa-spooler.cfg
```

4.10. Configure Zarafa Caldav

Zarafa Caldav is a component that enables users to view their calendar data by clients that support the Caldav standard, like Sunbird or Evolution. This component connects with the Zarafa Server using MAPI over HTTP.

Caldav and iCal push and retrieve complete calendars. Sunbird and other clients support both retrieving and pushing, while Evolution does only support retrieving of calendars.

The Zarafa Caldav component can be configured using a configuration file in the same fashion as the Zarafa Server. It supports both plain and SSL/TLS secured connections. To increase security it is recommended to enable secure Caldav connectivity exclusively.

The configuration options are:

```
server_bind
```

IP address to bind to. **`0.0.0.0`** for any address. Default value: **`0.0.0.0`**

```
ical_enable
```

Enable plain service with value **`yes`**. Default value: **`yes`**

```
ical_port
```

The plain service will listen on this port for incoming connections. Default Value: **`8080`**

```
icals_enable
```

Enable secure service with value **yes**. Default value: **no**

```
icals_port
```

The secure service will listen on this port for incoming connections. Default value: **8443**

```
server_socket
```

The http address of the Zarafa Server. Default value: **http://localhost:236/zarafa**

Important

It is not advised to specify the UNIX socket here. In default configuration the Zarafa Caldav will then be trusted by the **zarafa-server** (as set in its **local_admin_users** configuration setting). Unless Zarafa Caldav is specified to run as an untrusted user, it always authenticates users even if they provide no or wrong credentials!

```
ssl_private_key_file
```

The file that contains the private key used for encrypting the ssl connections. The absolute path to the file should be used. Default value: **/etc/zarafa/privkey.pem**

```
ssl_certificate_file
```

The file that contains the certificate for the server. The absolute path to the file should be used. Default value: **/etc/zarafa/cert.pem**

```
ssl_verify_client
```

Enable client certificate verification with value **yes**. Default value: **no**

```
ssl_verify_file / ssl_verify_path
```

The file or path to the files to verify the clients certificate with. The absolute path should be used for both options (no default).

```
[logging]
```

The Caldav component has the same configuration options as the server to configure logging options.

4.10.1. SSL/TLS

As mentioned before the Zarafa Caldav component supports SSL/TLS, for this the OpenSSL library is used.

The private key (for encryption) and the certificate (for authentication) file can be set in the configuration file with **ssl_private_key_file** and **ssl_certificate_file**.

The Zarafa Caldav component can also authenticate the calendar clients that try to connect to it verifying the client certificates using one or more verification files. This can be set with

ssl_verify_client, **ssl_verify_file** and **ssl_verify_path**. Certificates can be self-signed or signed by a trusted certificate authority.

The following command generates an RSA key of 2048 bytes:

```
openssl genrsa -out /etc/zarafa/privkey.pem 2048
```

This command creates a self-signed test certificate valid for 3 years:

```
openssl req -new -x509 -key /etc/zarafa/privkey.pem -out /etc/zarafa/cert.pem -days 1095
```

If a **.cer** file and a **.key** file are already present, you can create a **.pem** file from these using the following command:

```
cat my_server.key > my_server_combined.pem
cat my_server.cer >> my_server_combined.pem
```

And then use the **my_server_combined.pem** file for **ssl_private_key_file** or **ssl_certificate_file**. Please make sure first the **.key** file is processed, and then the **.cer** file.

4.10.2. Calendar access

Calendar folders served by the Zarafa Caldav component as accessed by URLs:

Tableau 4.2. CALDAV and iCal URLs

URL	Calendar
http://server:8080/ical/	user's own default calendar via ical (not recommended)
http://server:8080/caldav/	user's own default calendar
http://server:8080/caldav/<other-user>	Other-user's calendar
http://server:8080/caldav/<user>/<calendar>	user's self created subcalendar in a self created calendar
http://server:8080/caldav/public/<calendar>/	Calendar folder in the public folder.

Tableau 4.3. CALDAV and iCal URLs for MAC OSX iCal client

URL For MAC OSX iCal client	Calendar
http://server:8080/caldav/	User's calendar list
http://server:8080/caldav/<other-user>	Other-users calendar list
http://server:8080/caldav/public	Public folders list

Note

The **<other user>** or **<user>/<calendar>** is only reachable if the correct permissions are available. If you want to open another user's Calendar it is necessary to have *folder visible* permissions on the toplevel mailbox folder so the caldav servers can scan the mailbox for the right calendar folder. All other permissions are working the same as in Outlook.

Note

The Mac OS X iCal client is fully tested and supported up to 10.5.6. Additional information regarding client side setup can be found in the Zarafa User Manual.

4.11. Configure Zarafa Gateway (IMAP and POP3)

The Zarafa IMAP & POP3 Gateway enables users to view mail stored on the Zarafa Server with an IMAP or POP3 client. For example Mozilla Thunderbird or a mobile device with Microsoft Pocket Outlook. To access the user data, the Zarafa Gateway itself connects to the Zarafa Server with MAPI.

POP3 can only retrieve the mail in the Inbox from the server. IMAP on the other hand displays all folders that can contain mail, such as Drafts and Deleted Items. All sub-folders are shown as in Microsoft Office Outlook or the Zarafa WebAccess.

The Zarafa IMAP & POP3 Gateway can be configured with a configuration file. The configuration options are:

```
server_bind
```

IP address to bind to. **`0.0.0.0`** for any address. Default value: **`0.0.0.0`**

```
imap_enable
```

Enable IMAP service with value **yes**. Default value: **yes**

```
imap_port
```

The IMAP service will listen on this port for incoming connections. Default Value: **143**

```
imaps_enable
```

Enable secure IMAP service with value **yes**. Default value: **no**

```
imaps_port
```

The secure IMAP service will listen on this port for incoming connections. Default value: **993**

```
pop3_enable
```

Enable POP3 service with value **yes**. Default value: **yes**

```
pop3_port
```

The POP3 service will listen on this port for incoming connections. Default value: **110**

```
pop3s_enable
```

Enable secure POP3 service with value **yes**. Default value: **no**

```
pop3s_port
```

The secure POP3 service will listen on this port for incoming connections. Default value: **995**

```
imap_only_mailfolders
```

Enable only mailfolders to be shown with value **yes**. Default value: **yes**

```
server_socket
```

The http address of the Zarafa server. Default value: **http://localhost:236/zarafa**

Important

It is not advised to specify the UNIX socket here. In default configuration the Zarafa Gateway will then be trusted by the **zarafa-server** (as set in its **local_admin_users** configuration setting). Unless Zarafa Gateway is specified to run as an untrusted user, it always authenticates users even if they provide no or wrong credentials!

```
ssl_private_key_file
```

The file that contains the private key used for encrypting the ssl connections. The absolute path to the file should be used. Default value: **/etc/zarafa/privkey.pem**

```
ssl_certificate_file
```

The file that contains the certificate for the server. The absolute path to the file should be used. Default value: **/etc/zarafa/cert.pem**

```
ssl_verify_client
```

Enable client certificate verification with value **yes**. Default value: **no**

```
ssl_verify_file / ssl_verify_path
```

The file or path to the files to verify the clients certificate with. The absolute path should be used for both options (no default).

```
[logging]
```

The gateway has the same configuration options as the server to configure logging options.

4.11.1. SSL/TLS

The Zarafa Gateway supports SSL/TLS using the OpenSSL library. For more information see *Section 4.10.1, « SSL/TLS »*, as the options are exactly the same for these two components.

4.11.2. Important notes

IMAP and POP3 are provided for backward compatibility and will not provide the same experience like clients that support MAPI (Microsoft Outlook or our WebAccess). IMAP/POP3 clients use these protocols for mails only (where MAPI does mail, calendar and contacts).

Setting the Out of Office message is not possible with IMAP or POP3 clients.

Rules set in Microsoft Outlook do not work using the Zarafa IMAP & POP3 Gateway. Some clients can set rules but these rules are not related to the rules set by a MAPI enabled client.

Deleting a mail using IMAP will mark the mail for deletion. This is not shown in Microsoft Outlook and Zarafa WebAccess. The mail will be deleted when the client expunges the folder. Some clients allow to expunge folders manually and some have settings when to expunge a folder. Other clients expunge the folder automatically when a mail is deleted.

Moving mail to a different folder with IMAP is done by copying the mail to the new folder and mark the originating mail for deletion. As long as the the original mail is not expunged from its folder, the mail will be shown in both folders as stated above.

4.12. Configure Zarafa Quota Manager

Users can collect a lot of email, while disk space can be limited. The Zarafa Quota Manager can be used to set server-wide or user specific space quotas. The Zarafa Quota Manager knows three levels: warn, soft and hard quota. When one of the levels will be reached, the user receives an email with the quota sizes and which quota level was reached.

The quota settings can be configured server-wide in the **`server.cfg`** or per user via the user plugin.

When a user reaches the warning quota level, the user will receive an email with a warning and quota information. As the user reaches the soft quota limit, the user will not be able to sent email until the size of the store is reduced. When the hard quota limit is reached, email can also not be delivered to that user anymore.

4.12.1. Setup server-wide quota

The server-wide quota can be configured in the configuration file of the server:

```
quota_warn = 100
quota_soft = 150
quota_hard = 200
```

The values are all in megabytes. These values will be honored for all users present in the server. When the values are set to **`0`**, that particular quota level is disabled.

4.12.2. Setup quota per user

By using the **`zarafa-admin`** tool, the user quota can be set for a specific user. Example:

Set the quota of the user John with the settings: Warning level to 80 Mb, soft level to 90 Mb and hard level to 100 Mb.

```
zarafa-admin  -u john  --qo 1  --qw 80  --qs 90  --qh 100
```

Note

Set user quota with **`zarafa-admin`** does not work with LDAP. With LDAP the properties are stored in the LDAP server per user. See the *Chapitre 8, Gestion des utilisateurs* for more information.

4.12.3. Monitoring for quota exceeding

The zarafa-monitor program checks every hour (by default) for users who have exceeded a quota level and sends emails to a user when the warning or soft quota limit is exceeded. Global quota settings can be set in the server configuration. User specific levels can be set via **`zarafa-admin`**

when using the db or unix plugin, or by editing the LDAP values as described in the User Management section.

To start the zarafa-monitor, use:

```
/etc/init.d/zarafa-monitor start
```

or

```
zarafa-monitor -c /etc/zarafa/monitor.cfg
```

The **zarafa-monitor** will daemonise, so the prompt will almost immediately return. Use **-F** to start it in the foreground. More information about the configuration options can be found in the manual page:

```
man zarafa-monitor.cfg
```

4.12.4. Quota warning templates

When working with the zarafa-monitor, it is possible to modify the contents of the email which will be sent out when a user or company exceeds its quota. For each quota level a separate quota template can be specified, these can be configured with the following options:

- **userquota_warning_template**
- **userquota_soft_template**
- **userquota_hard_template**
- **companyquota_warning_template**

By default the templates are stored in **/etc/zarafa/quotamail**, in each of these templates certain variables are provided which will be substituted for the real value before the email is sent:

- **ZARAFA_QUOTA_NAME** - The name of the user or company who exceeded his quota
- **ZARAFA_QUOTA_COMPANY** - The name of the company to which the user belongs
- **ZARAFA_QUOTA_STORE_SIZE** - When a user exceeds his quota, this variable contains the total size of the user's store. When a company exceeds its quota this variable contains the total size of all stores, including the public store within the company space.
- **ZARAFA_QUOTA_WARN_SIZE** - The quota warning limit for the user or company.
- **ZARAFA_QUOTA_SOFT_SIZE** - The quota soft limit for the user or company.
- **ZARAFA_QUOTA_HARD_SIZE** - The quota hard limit for the user or company.

Note

Variables containing a size always include the size unit (**B**,**KB**,**MB**,**GB**) as part of the variable.

4.13. Configure Zarafa Search

The **zarafa-search** service, introduced in ZCP 7.10, offers full text searching capabilities for the Zarafa Server. The service will continuously index all mails, and optionally their attachments, of a

single zarafa-server instance. Each zarafa-server instance in a multi-server setup needs it's own zarafa-search service.

When searching for a particular mail, the required time to find the requested emails will be seriously reduced. When attachment indexing is enabled, it is even possible to index the contents of attached files (for common file types that contain text).

4.13.1. Enabling the search service

To start the indexing service execute the following command:

```
/etc/init.d/zarafa-search start
```

To enable the full-text searching, edit the **/etc/zarafa/server.cfg** configuration file:

```
search_enabled = yes
```

During searching the zarafa-server will connect with the **zarafa-search** service. To set the connection path change the following configuration option:

```
search_socket = file://var/run/zarafa-search
```

4.13.2. Search configuration

During indexing, the index file for each store is stored on the harddisk. The location of these files can be configured in **/etc/zarafa/search.cfg**:

```
index_path          =   /var/lib/zarafa/index/
```

In this folder a file will be created for each store located on the Zarafa server node. A state file will also be present to remember where the indexing process has left upon restart.

Important

The files within this index path should not be touched while the indexer is running. If a store must be re-indexed, the **zarafa-search** must be stopped first before deleting the file for that particular store.

The **zarafa-search** service uses streaming synchronization offered by the zarafa-server for fast indexing of messages. To enable streaming, ensure that the following configuration option is enabled in the zarafa-server config:

```
enable_enhanced_ics = yes
```

This option is enabled by default, and normally there is no reason to disable it.

4.13.3. Attachments

Optionally the contents of attachments can be indexed as well. When this is enabled, searching for a message will also search through the attachment text as well.

To enable indexing of attachments can be done in **/etc/zarafa/search.cfg**:

```
index_attachments   = yes
```

Indexing of attachments is done through parsing the attachments to plain text and indexing the text into the main index for the email. The required time to parse and index a particular attachment depends on the actual size of the attachment. To prevent large attachments adding latency to the total indexing time, the configuration option **index_attachment_max_size** can be used to prevent large attachments to be indexed. The value provided to this configuration option must be set in kilobytes.

To parse the attachments to plain text a separate configuration script must be provided. By default this script is installed to **/etc/zarafa/searchscripts/attachments_parser** but the exact location can be configured using the configuration option **index_attachment_parser**.

The default script **attachments_parser** will use the file **attachments_parser.db** to decide how the attachment should be parsed to plain text. Within this file is a list containing the command to parse each attachment type to plain text. This file can be edited to control the way attachments are parsed and to add or remove support for particular attachment types.

The layout of each line is as followed:

```
<mime-type>;<extension>     `<command>`
```

Each line can have as many mime-types and extensions as needed, each mime-type and extension must be separated using semi-columns. The command must read **/dev/stdin** for the attachment data and must return the plain text through **/dev/stdout**. Some tools cannot parse attachment data from a stream, and require the data to be provided as file. To store the attachment in a temporary file, the script **zmktemp** can be used. That script will write all attachment data in a temporary file and print the location of the file to **/dev/stdout**.

Attachments which cannot be parsed (for example images), the command **echo -n** can be used.

After editing the command, it is advisable to test it to see if the desired output is returned. Testing the command can be done by executing the following command on the command line:

```
cat <attachment> | <command>
```

The resources used by the **attachments_parser** during the parsing of a single attachment can be restricted by limiting the total memory and CPU time usage. To control the maximum amount of memory the script can use is controlled by the configuration option **index_attachment_parser_max_memory**. By default this value is set to **0**, to disable any memory consumption restriction. If a restriction should be applied, the maximum number of bytes should be provided. The best restriction size depends on the maximum attachment size which can be provided to the script (configured using **index_attachment_max_size**) and the 3rd party tools used to parse the attachments.

To prevent the script to take too much time, the configuration option **index_attachment_parser_max_cputime** can be used. By default this value is set to **0**, to disable any CPU time restriction. If a restriction should be applied, the maximum number of seconds should be provided. The best restriction depends on the 3rd party tools used to parse the attachments.

If either of these limits is exceeded the script will be canceled and the attachment will not be indexed.

Configuration des composants tierces

5.1. Configuration du serveur Web

Normalement, le package Zarafa configure automatiquement PHP sur votre système. Dans la plupart des cas, ce chapitre peut être ignoré pour aller directement à la *Section 5.1.2, « Configuration Apache »*.

5.1.1. Configuration PHP

PHP est indispensable à l'utilisation de WebAccess. L'extension PHP-MAPI est installée dans le répertoire par défaut de la distribution :

- Red Hat Enterprise Linux: **`/usr/lib/php5/modules/`**
- SLES / OpenSUSE: **`/usr/lib/php/extensions/`**
- Debian: **`/usr/lib/php5/20060613/`**
- Ubuntu: **`/usr/lib/php5/20060613/`**

Si un répertoire différent a été sélectionné pour les extensions PHP, il faudra déplacer les fichiers **`mapi.so*`** vers cet emplacement, p. ex. :

```
mv /usr/lib/php/mapi.so* \
      /usr/local/lib/php/
```

Pour trouver l'emplacement des extensions PHP, veuillez exécuter la commande suivante :

```
php-config --extension-dir
```

Une fois que l'extension PHP-MAPI est dans le répertoire adéquat, il faut l'ajouter au fichier de configuration **`php.ini`**. Si ce n'est pas déjà fait, ajouter la ligne suivante au fichier php.ini :

```
extension = mapi.so
```

Les emplacements habituels pour le fichier**`php.ini`** sont généralement :

```
/etc/php.ini
```

```
/etc/php5/apache2/php.ini
```

À l'aide de la fonction **`phpinfo()`** il est possible de vérifier si le module se charge correctement. Effectuer une recherche sur le mot clé ‘MAPI’ pour procéder à la vérification du module. L'affichage de phpinfo peut également être effectué en exécutant **`php -i`** en ligne de commande si **`php cli`** est installé.

5.1.2. Configuration Apache

Pour charger correctement l'extension mapi.so récemment installée, le serveur Web doit être redémarré. L'exemple suivant montre comment redémarrer Apache2:

```
/etc/init.d/apache2 restart
```

ou

```
/etc/init.d/httpd restart
```

5.1.2.1. For WebAccess

Les fichiers du client Web sont installés par défaut dans le répertoire de WebAccess. Il faut s'assurer de pouvoir accéder à la page d'authentification du client Web en se rendant sur l'URL adéquate :

```
http://<ip-address server>/webaccess/
```

Si la page d'authentification ne s'affiche pas, le serveur Web devra être reconfiguré afin de permettre l'accès au répertoire adéquat. L'exemple suivant présente une configuration pour Apache2:

```
Alias /webaccess /usr/share/zarafa-webaccess/
<Directory /usr/share/zarafa-webaccess/>
        AllowOverride None
        Order allow,deny
        Allow from all
</Directory>
```

Il faut s'assurer que le répertoire correct contenant les fichiers PHP de WebAccess a bien été saisi. La commande suivante permettra à Apache2 de recharger sa configuration :

```
/etc/init.d/apache2 reload
```

WebAccess devrait alors s'afficher. Si ce n'est toujours pas le cas, veuillez consulter la *Section 2.3, « Résolution de problèmes d'installation »* pour plus d'information.

5.1.2.2. For WebApp

The website files are by default installed in the WebApp directory. Make sure the webclient's login page can be opened by browsing to the correct url:

```
http://<ip-address server>/webapp/
```

Si la page d'authentification ne s'affiche pas, le serveur Web devra être reconfiguré afin de permettre l'accès au répertoire adéquat. L'exemple suivant présente une configuration pour Apache2:

```
Alias /webapp /usr/share/zarafa-webapp/
<Directory /usr/share/zarafa-webapp/>
        AllowOverride None
        Order allow,deny
        Allow from all
</Directory>
```

Make sure the correct directory holding the PHP WebApp files is typed. The following command will tell apache2 to reread its config file:

```
/etc/init.d/apache2 reload
```

The WebApp should now be visible. If it still does not show up, please see *Section 2.3, « Résolution de problèmes d'installation »* for more information.

When leaving the configuration at this point, Apache will request the browsers to cache all files as long as they see fit. This may mean that users are still seeing the old interface while the WebApp package on the server has been upgraded. To fix this, the package comes with an example configuration that includes instructions to the browsers on how long WebApp resources may be kept around.

Using this, we are saying that Javascript and CSS files need to be checked against the server versions very often, but Apache can serve these files very quickly from the filesystem. For images, we allow the clients to keep using them for a much longer period (2 months). For this, we use the FileETag setting of Apache to generate a unique identifier for each served static file. To use this, the Apache modules mod_expires and mod_headers need to be loaded.

The following can be included in the Apache configuration within the <Directory> directive as described above:

```
FileETag All

ExpiresActive On

<filesMatch "\.(jpg|gif|png)$">
    ExpiresDefault "access plus 2 months"
    Header append Cache-Control "public"
</filesMatch>

<FilesMatch "\.(js|css)$">
    ExpiresDefault "access plus 2 weeks"
    Header append Cache-Control "no-cache, must-revalidate"
</FilesMatch>

<filesMatch "\.(php)$">
    ExpiresActive Off
    Header set Cache-Control "private, no-cache, no-store, proxy-revalidate, no-transform"
    Header set Pragma "no-cache"
</filesMatch>
```

The example zarafa-webapp.conf that comes with the WebApp package contains a more extensive version of this. Especially if you have a lot of users with Internet Explorer, this will be better suited for you than the terse example above.

5.1.3. Apache comme proxy HTTP

Les données transmises entre le client et le serveur sont compressées en XML et contenues dans des paquets HTTP. L'utilisation de HTTP permet aux paquets d'être transférés par un proxy (ou un serveur Web intégrant la fonctionnalité proxy, par exemple Apache version 2).

Les lignes suivantes sont un exemple de configuration possible sur Apache, afin de transférer les connexions entrantes sur le port**`80`** vers le serveur Zarafa sur le port **`236`**. Si de plus, le serveur Apache accepte les connexions HTTPS, alors les connexions par proxy peuvent également être chiffrées. Les modules**`proxy`** et **`proxy_html`** d'Apache doivent être chargés.

```
<IfModule mod_proxy.c>
       ProxyPass /zarafa http://127.0.0.1:236/
       ProxyPassReverse /zarafa http://127.0.0.1:236/
</IfModule>
```

Cela signifie que les URLs qui commencent par **`/zarafa`** seront transférées à **`localhost`** sur le port **`236`**, où le serveur Zarafa est à l'écoute des connexions entrantes. Ces lignes peuvent être placées globalement ou bien dans une déclaration VirtualHost.

Note

Il ne faut pas oublier que l'utilisation d'un proxy HTTP a des conséquences sur les performances, c'est pourquoi il n'est pas recommandé d'en faire l'usage dans les infrastructures de grande taille.

5.2. Configuration de l'intégration ZCP OpenLDAP

Dans nombre d'infrastructures réseau, OpenLDAP est employé comme serveur d'annuaire pour contrôler un certain nombre de données dont les plus notables sont les utilisateurs et leurs autorisations. ZCP s'intègre avec les serveurs LDAP, et prend notamment en charge OpenLDAP.

Zarafa doesn't include a LDAP server in the product, so if there's not yet a LDAP server available in the environment, one has to be setup or the non-LDAP user plugin has to be used. Please read the documentation of the used Linux distribution on how to setup an OpenLDAP server. Zarafa provides an example LDIF file in *Chapitre 14, Appendix C: Example LDIF*.

Les connexions au serveur OpenLDAP s'effectuent sur le port **389** ou **636** (SSL). Pour une meilleure rapidité et fiabilité, il est toujours préférable d'installer OpenLDAP sur le même hôte physique que le serveur Zarafa qui réplique les données du serveur LDAP principal. En plus des améliorations de performance, cela permet également au serveur Zarafa de continuer à fonctionner même si le serveur LDAP principal s'arrête.

La configuration sera expliquée dans les paragraphes suivants. Vérifier l'emplacement des fichiers de configuration avant d'effectuer toute modification.

La configuration OpenLDAP est généralement située dans le répertoire **`/etc`**, selon la distribution employée :

- Red Hat Enterprise Linux: **`/etc/openldap`**
- SUSE: **`/etc/openldap`**
- Debian & Ubuntu: **`/etc/ldap`**

Dans ce guide, nous utiliserons : **`/etc/openldap`**

5.2.1. Configurer OpenLDAP afin d'utiliser les schémas Zarafa

Pour configurer OpenLDAP afin d'utiliser les schémas LDAP de Zarafa, la directive de configuration suivante doit être ajoutée à **`/etc/openldap/slapd.conf`**:

```
include         /etc/openldap/schema/zarafa.schema
```

Copiez ensuite le fichier de schéma vers le répertoire LDAP :

```
cp /usr/share/doc/zarafa/zarafa.schema /etc/openldap/schema/zarafa.schema
```

Note

La plupart des distributions Linux récentes utilisent OpenLDAP en mode de configuration dynamique. Pour plus d'information sur l'installation du schéma Zarafa pour un serveur OpenLDAP doté d'une configuration dynamique, veuillez consulter *http://www.zarafa.com/wiki/index.php/OpenLdap:_Switch_to_dynamic_config_backend_%28cn%3Dconfig%29*.

5.2.2. LDAP indices

Indexing entries is a way to improve performance when a Zarafa Server performs a filtered search on the LDAP directory. The following table show the most important attributes to index and the type of index that should be implemented.

Tableau 5.1. LDAP indices

Attribute name	Type
cn	pres,eq,sub
gidNumber	pres,eq
mail	pres,eq,sub
memberUid	pres,eq
objectClass	pres,eq
ou	pres,eq
sn	pres,eq,sub
uid	pres,eq
uidNumber	pres,eq
zarafaAliases	pres,eq,sub
zarafaAccount	pres,eq
zarafaSendAsPrivilege	preq,eq
zarafaViewPrivilege	pres,eq

Depending on the Zarafa ldap configuration the attributes may be different. Please check the openldap or syslog logfiles for attributes which are not yet indexed, see example below:

```
May 13 14:37:17 zarafa slapd[4507]: <= bdb_equality_candidates: (mail) not indexed
```

The reported attributes should be added as indices to OpenLDAP configuration.

5.2.3. Configurer ZCP pour OpenLDAP

Pour intégrer ZCP avec un serveur OpenLDAP, modifier l'option suivante dans le fichier de configuration **`ldap.cfg`** :

Définir l'adresse IP ou le nom d'hôte du serveur LDAP dans l'option **`ldap_host`**.

```
ldap_host = localhost
```

Par défaut, le protocole en clair de LDAP sera utilisé. Pour configurer LDAP en mode sécurisé, modifier les paramètres suivants. Un tutoriel de configuration OpenLDAP avec certificats SSL est disponible sur *http://wiki.zarafa.com*.

```
ldap_port = 389
ldap_protocol = ldap
```

To connect ZCP to multiple LDAP servers, use the following setting:

```
ldap_uri = ldap://ldapserver1:389 ldap://ldapserver2:389
```

The different ldap uri's should be seperated by a whitespace. When using the **`ldap_uri`** option, the options **`ldap_host`**, **`ldap_port`** and **`ldap_protocol`** are ignored.

Le serveur Zarafa lira uniquement les données à partir du serveur OpenLDAP L'utilisateur bind spécifié doit au minimum posséder l'accès en lecture sur le serveur LDAP.

```
ldap_bind_user = cn=Manager,dc=example,dc=com
ldap_bind_passwd = secret
ldap_authentication_method = bind
```

La méthode d'authentification peut être définie sur **`password`**, afin que le serveur Zarafa compare le mot de passe chiffré du serveur LDAP avec le mot de passe chiffré fournit par l'utilisateur lors de son identification.

Pour cette méthode, l'utilisateur bind spécifié doit avoir le rang d'administrateur dans OpenLDAP et doit posséder l'accès en lecture sur l'attribut du mot de passe.

La base de recherche LDAP (base DN) à partir de laquelle la recherche des différents objets devra démarrer. Ce doit être la 'racine' du dossier LDAP contenant les utilisateurs, les groupes et les contacts.

```
ldap_search_base = dc=example,dc=com
ldap_object_type_attribute = objectClass
ldap_user_type_attribute_value = posixAccount
ldap_group_type_attribute_value = posixGroup
ldap_contact_type_attribute_value = zarafa-contact
ldap_company_type_attribute_value = zarafa-company
ldap_addresslist_type_attribute_value = zarafa-addresslist
ldap_dynamicgroup_type_attribute_value = zarafa-dynamicgroup
```

Based on the ldap_object_type attribute the Zarafa Server will create an object in the MySQL database, so it's get listed in the Global Address Book. Make sure that the values are always unique for one type of object, as Zarafa needs to be able to distinguish the different objects.

5.2.4. Configuration des utilisateurs

Normally a user store is created for each object in the LDAP directory that has the user type attribute as mentioned in the previous section (posixAccount in the previous example). An additional search filter can be specified to limit store creation to a subset of the objects that have the user type attribute. For example:

```
ldap_user_search_filter = (zarafaAccount=1)
```

Tous les champs relatifs aux utilisateurs peuvent être reliés à l'aide des options suivantes :

```
ldap_user_unique_attribute = uidNumber
ldap_user_unique_attribute_type = text
```

```
ldap_fullname_attribute = cn
ldap_loginname_attribute = uid
ldap_emailaddress_attribute = mail
ldap_emailaliases_attribute = zarafaAliases
ldap_password_attribute = userPassword
ldap_isadmin_attribute = zarafaAdmin
ldap_nonactive_attribute = zarafaSharedStoreOnly
```

L'attribut unique de l'utilisateur (ldap_user_unique_attribute) est le lien entre une boîte aux lettres dans la base de données et l'utilisateur correspondant dans LDAP. Il faut s'assurer que ce champ reste inchangé car le serveur Zarafa interprèterait un tel changement comme une suppression (puis création) d'utilisateur et rendrait alors orpheline la base de stockage correspondante.

The email aliases are shown in the Global Address Book details and can be used for resolving email aliases in Postfix. However it is not possible to deliver email to email aliases with the dagent directly, this needs to be resolved by Postfix.

Les informations supplémentaires telles que les adresses, numéros de téléphone ou sociétés peuvent être liées à l'aide d'un fichier de configuration supplémentaire :

```
!propmap /etc/zarafa/ldap.propmap.cfg
```

Les attributs spécifiques aux utilisateurs peuvent également être employés pour les contacts.

5.2.5. Configuration des groupe

Les groupes peuvent également être filtrés par un filtre de recherche supplémentaire.

```
ldap_group_search_filter = (objectClass=zarafa-group)
ldap_group_unique_attribute = gidNumber
ldap_group_unique_attribute_type = text
```

Pour les relations d'affiliation entre les groupes et les utilisateurs, chaque objet de groupe possède un attribut d'adhésion au groupe Celui ci peut être configuré de la façon suivante :

```
ldap_groupmembers_attribute = memberUid
```

Le serveur Zarafa utilisera par défaut l'attribut unique de l'utilisateur comme valeur pour l'attribut d'adhésion au groupe (ldap_groupmembers_attribute). Ceci peut être modifié par l'attribut de relation d'adhésion au groupe.

```
ldap_groupmembers_attribute_type = text
ldap_groupmembers_relation_attribute = uid
```

Les groupes peuvent être définis comme groupes de sécurité à l'aide de l'attribut groupe de sécurité. Les groupes de sécurité sont disponibles dans le carnet d'adresses global lors de la création d'un nouveau courriel ou lors du paramétrage de permissions. Pour le permettre, l'attribut (dans ce cas **`zarafaSecurityGroup`**) doit être paramétré avec la valeur **`1`**. Lorsque l'attribut zarafaSecurityGroup est paramétré avec la valeur 0, le groupe sera défini comme simple groupe de diffusion. Les groupes de diffusion ne sont disponibles dans le carnet d'adresses global que lors de la création d'un nouveau courriel mais ils ne peuvent pas être utilisés pour définir les permissions des boîtes aux lettres.

```
ldap_group_security_attribute = zarafaSecurityGroup
```

```
ldap_group_security_attribute_type = boolean
```

5.2.6. Configuration des listes d'adresses

Les listes d'adresses sont des groupes d'utilisateur qui répondent à des critères personnalisées. Ces listes d'adresses sont affichées dans des sous-répertoires du carnet d'adresses global.

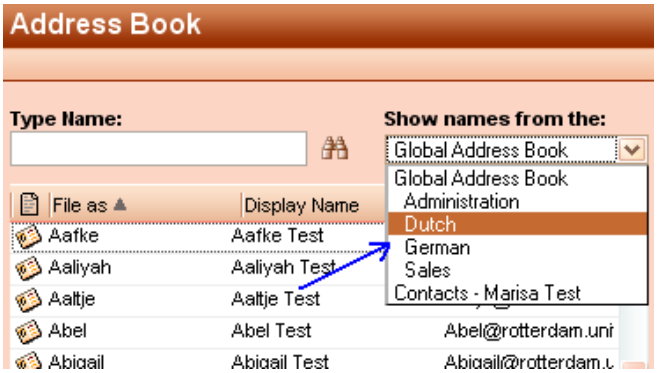


Figure 5.1. Listes d'adresses dans le carnet d'adresses global

Modifier ou ajouter dans le fichier de configuration **`ldap.cfg`** les paramétrages d'objet de listes d'adresses suivants :

```
ldap_addresslist_search_filter =
ldap_addresslist_unique_attribute = gidNumber
ldap_addresslist_unique_attribute_type = text
ldap_addresslist_filter_attribute = zarafaFilter
ldap_addresslist_name_attribute = cn
```

Consulter la *Section 8.5, « Gestion des utilisateurs avec LDAP ou Active Directory »* pour plus d'information sure la gestion des listes d'adresses.

5.2.7. Vérifier la configuration LDAP

Une fois la configuration LDAP achevée, les modifications seront activées en redémarrant le serveur Zarafa.

```
/etc/init.d/zarafa-server reload
```

Pour vérifier si les utilisateurs et les groupes sont correctement répertoriés dans Zarafa par l'annuaire LDAP, veuillez exécuter la commande suivante :

```
zarafa-admin -l
```

pour les utilisateurs, tandis que pour les groupes :

```
zarafa-admin -L
```

Si aucun utilisateur ni groupe n'est affiché, veuillez consulter les messages d'erreur du fichier de journalisation du serveur Zarafa. Définir le niveau de journalisation **`log_level`** sur **`6`** dans le fichier de configuration **`/etc/zarafa/server.cfg`** affichera l'historique de toutes les requêtes LDAP ainsi que toutes les erreurs possibles.

Note

La première fois que **`zarafa-admin -l`** est exécuté, toutes les boîtes aux lettres seront créées. Ceci peut prendre un certain temps et demander de la patience.

Des informations supplémentaires sur les autres attributs LDAP disponibles sont contenues dans la page man.

```
man zarafa-ldap.cfg
```

5.3. Configuration de l'intégration ZCP Active Directory

5.3.1. Installation du plugin Zarafa ADS Plugin et des fichiers schémas

ZCP offre un installeur qui apporte une extension au schéma Active Directory et qui fournit un snap-in Active Directory afin de gérer les attributs spécifiques à Zarafa.

Le plugin Zarafa ADS n'est disponible que pour les éditions commerciales de ZCP et peut être téléchargé sur *https://portal.zarafa.com*.

Le plugin Zarafa ADS doit être installé en tant qu'administrateur local sur le serveur Active Directory qui assure le rôle de maître de schéma.

5.3.1.1. Windows 2000 Server

Lorsque l'installation est effectuée sur un serveur Windows 2000, la configuration nécessite un accès en écriture afin d'actualiser le schéma Active Directory. Pour obtenir l'accès en écriture, la clé du registre "Schema Update Allowed" doit être activée.

Pour modifier la clé de registre, veuillez suivre les étapes suivantes :

1. Cliquer sur Démarrer, cliquer sur Exécuter, puis dans la boîte Ouvrir, saisir : **`regedit`**, puis appuyer sur Entrée.
2. Retrouver puis cliquer sur la clé de registre suivante :

   ```
   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Parameters
   ```

3. Dans le menu Édition, cliquer sur Nouveau, puis cliquer sur la valeur **`DWORD`**.
4. Saisir la valeur de 'Value Data' lorsque la valeur du registre suivante s'affiche :

   ```
   Value Name: Schema Update Allowed
   Data Type: REG_DWORD
   Base: Binary
   Value Data: Type 1 to enable this feature, or 0 (zero) to disable it.
   ```

5. Quitter l'éditeur de registre.

Maintenant l'installeur Zarafa Active Directory peut être exécuté. Pour plus d'information, veuillez consulter : *http://support.microsoft.com/kb/285172*

Note

Ne pas oublier de réactiver la clé de registre une fois l'installation achevée.

5.3.1.2. Windows 2003/2008 Server

Pour Windows 2003 et 2008 Server, il est possible de progresser à chaque étape de l'installation en cliquant sur le bouton Suivant.

Si le plugin Zarafa ADS est installé, les attributs Zarafa pourront être modifiés. Pour modifier un utilisateur, aller sur **`Utilisateurs et Ordinateurs`**, sélectionner un utilisateur puis ses propriétés. L'onglet Zarafa devrait s'afficher si l'installation s'est déroulée correctement.

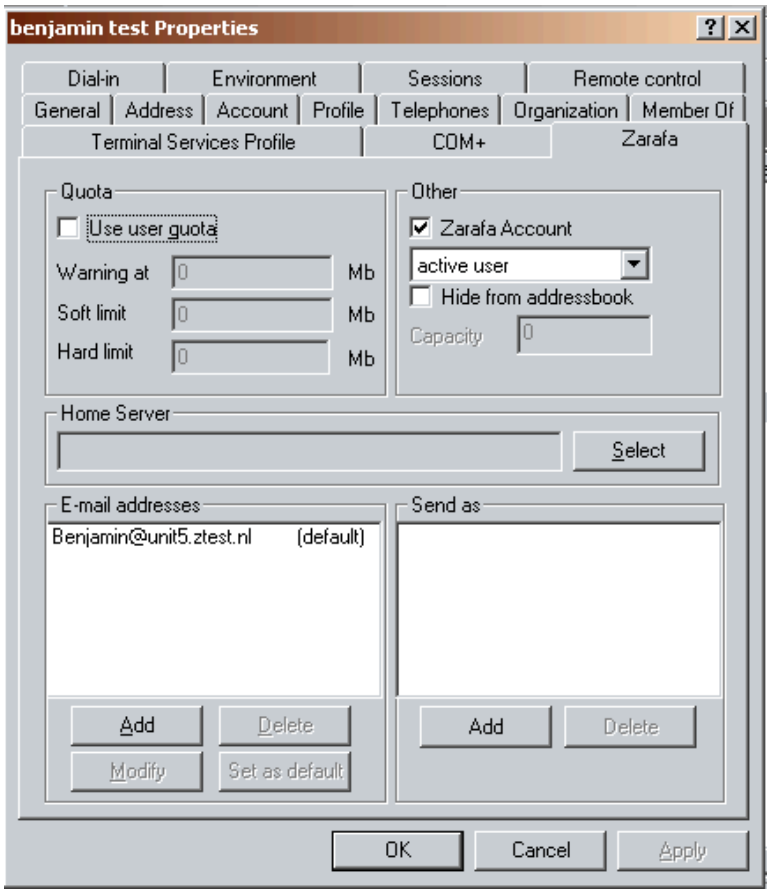


Figure 5.2. Onglet de l'utilisateur Zarafa

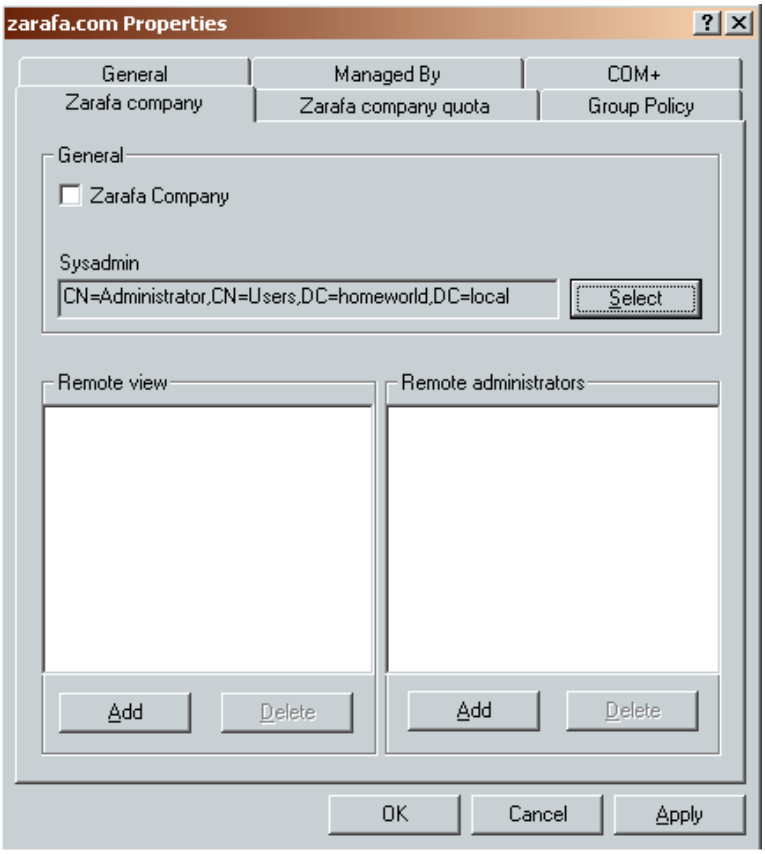


Figure 5.3. Onglet du groupe Zarafa

5.3.2. Configurer ZCP pour ADS

Pour intégrer ZCP avec un serveur d'annuaire Active Directory, modifier l'option suivante dans le fichier de configuration **`ldap.cfg`** :

Définir l'adresse IP ou le nom d'hôte du serveur Active Directory dans l'option ldap_host.

```
ldap_host = 192.168.0.100
```

Par défaut, le protocole en clair de LDAP sera utilisé. Pour configurer LDAP en mode sécurisé, modifier les paramètres suivants :

```
ldap_port = 636
ldap_protocol = ldaps
```

Un tutoriel de configuration Active Directory avec certificats SSL est disponible sur *un article de notre wiki*[1].

To connect ZCP to multiple Active Directory servers, use the following setting:

```
ldap_uri = ldap://dc1:389 ldap://dc2:389
```

The different ldap uri's should be seperated by a whitespace. When using the **`ldap_uri`** option, the options **`ldap_host`**, **`ldap_port`** and **`ldap_protocol`** are ignored.

[1] http://www.zarafa.com/wiki/index.php/Configure_Active_Directory_with_SSL

Le serveur Zarafa lira uniquement les données à partir du (et n'écrira jamais vers le) serveur LDAP ou Active Directory. C'est pourquoi l'utilisateur bind spécifié doit au minimum posséder l'accès en lecture sur le serveur LDAP.

```
ldap_bind_user = cn=administrator,cn=users,dc=example,dc=com
ldap_bind_passwd = secret
ldap_authentication_method = bind
```

The LDAP search base (base DN) specifies a branch that the Zarafa Server with limit itself to. Ce doit être la 'racine' du dossier LDAP contenant les utilisateurs, les groupes et les contacts.

```
ldap_search_base = dc=example,dc=com
```

By the following type attributes the Zarafa Server knows what objects to create in the database and what to list in the Global Address Book. Make sure these values are all unique.

```
ldap_object_type_attribute = objectClass
ldap_user_type_attribute_value = User
ldap_group_type_attribute_value = Group
ldap_contact_type_attribute_value = Contact
ldap_company_type_attribute_value = ou
ldap_addresslist_type_attribute_value = zarafa-addresslist
ldap_dynamicgroup_type_attribute_value = zarafa-dynamicgroup
```

As performance optimization feature the setting **ldap_page_size** was implemented to limit result sets in pages of this size downloading fewer results at a time from the LDAP server.

```
# Default ADS MaxPageSize is 1000.
ldap_page_size = 1000
```

5.3.3. Configuration des utilisateurs

```
which have specified user type attribute an additional search filter can be specified. Par
 exemple :
```

```
ldap_user_search_filter = (zarafaAccount=1)
```

Tous les champs relatifs aux utilisateurs peuvent être reliés à l'aide des options suivantes :

```
ldap_user_unique_attribute = objectGUID
ldap_user_unique_attribute_type = binary
```

```
ldap_fullname_attribute = cn
ldap_loginname_attribute = sAMAccountName
ldap_emailaddress_attribute = mail
ldap_emailaliases_attribute = otherMailbox
ldap_password_attribute =
ldap_isadmin_attribute = zarafaAdmin
ldap_nonactive_attribute = zarafaSharedStoreOnly
```

The unique user attribute is the mapping between a mailbox in the database and the actual user. Make sure this field can never be changed, otherwise a user deletion will be triggered by the Zarafa Server.

The email aliases are shown in the Global Address Book details and can be used for email aliases in Postfix. However it’s not possible to deliver email to email aliases.

Les informations supplémentaires telles que les adresses, numéros de téléphone ou sociétés peuvent être liées à l'aide d'un fichier de configuration supplémentaire :

```
!include /etc/zarafa/ldap.propname.cfg
```

The specified attributes for users will also be used for the contacts.

Important

The attribute **`otherMailbox`** is by default not indexed in Active Directory. It's required to index this attribute in Active Directory, otherwise the Active Directory server will have a high CPU load during search queries on this attribute. For more information about indexing attributes in Active Directory, see *http://go.microsoft.com/fwlink/?LinkId=46790*.

5.3.4. Configuration des groupe

The groups can be as well filtered by an extra search filter.

```
ldap_group_search_filter =
ldap_group_unique_attribute = objectSid
ldap_group_unique_attribute_type = binary
```

Pour les relations d'affiliation entre les groupes et les utilisateurs, chaque objet de groupe possède un attribut d'adhésion au groupe Celui ci peut être configuré de la façon suivante :

```
ldap_groupmembers_attribute = member
ldap_groupmembers_attribute_type = dn
```

By the security group attribute group can be specified as security groups in Active Directory.

Security groups will only displayed when settings permissions and are not default available in the Global Address Book.

```
ldap_group_security_attribute = groupType
ldap_group_security_attribute_type = ads
```

5.3.5. Configuration des listes d'adresses

Les listes d'adresses sont des groupes d'utilisateur qui répondent à des critères personnalisées. These addresslists are showed as subfolders of the Global Address Book.

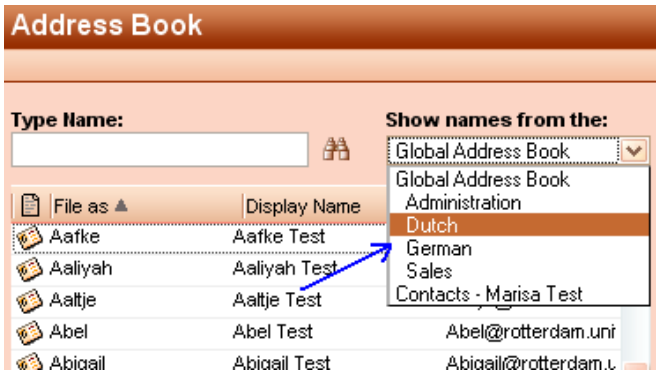


Figure 5.4. Listes d'adresses dans le carnet d'adresses global

Change or add in **ldap.cfg** the following configuration settings for the addresslist objects.

```
ldap_addresslist_search_filter =
ldap_addresslist_unique_attribute = cn
ldap_addresslist_unique_attribute_type = text
ldap_addresslist_filter_attribute = zarafaFilter
ldap_addresslist_name_attribute = cn
```

See the *Section 8.5, « Gestion des utilisateurs avec LDAP ou Active Directory »* for more information on how to administer address lists.

5.3.6. Testing Active Directory configuration

Une fois la configuration LDAP achevée, les modifications seront activées en redémarrant le serveur Zarafa.

```
/etc/init.d/zarafa-server reload
```

To test users and groups will be listed, use:

```
zarafa-admin -l
```

et

```
zarafa-admin -L
```

Si aucun utilisateur ni groupe n'est affiché, veuillez consulter les messages d'erreur du fichier de journalisation du serveur Zarafa. Définir le niveau de journalisation sur **6** dans le fichier de configuration **/etc/zarafa/server.cfg** affichera l'historique de toutes les requêtes LDAP et de toutes les erreurs possibles.

La première fois que **zarafa-admin -l** est exécuté, toutes les boîtes aux lettres seront créées. Ceci peut prendre un certain temps et demander de la patience.

More information about the other available LDAP attributes can be found in the man page.

```
man zarafa-ldap.cfg
```

See *Chapitre 8, Gestion des utilisateurs* for Zarafa user management with Active Directory.

5.4. ZCP Postfix integration

ZCP does not include it’s own MTA, but can be integrated all established MTAs found in modern Linux distributions. Although ZCP support most Linux MTAs, we advise to use Postfix.

In order to deliver an email into a user’s mailbox, the **zarafa-dagent** is executed. Messages are passed to the **zarafa-dagent** from the standard input or by the LMTP protocol. The usage of LMTP is the recommended delivery method as this enable the Single Instance Attachment Storage.

A few examples of the ZCP Postfix integration are described in the following sections. Keep in mind that Postfix is very flexible, so many different configurations are possible, most of which are beyond the scope of this document.

Note

Configuring antispam and antivirus scanning is beyond the scope for this manual. On the internet many example configurations are available for the most common MTAs and scanners.

5.4.1. Configure ZCP Postfix integration with OpenLDAP

The Postfix MTA can connect to an OpenLDAP server to resolve primary mail addresses and aliases of users and groups. The Postfix package in most Linux distributions has LDAP support enabled by default. To read more about Postfix LDAP support see *the LDAP README*[2] on the Postfix website.

All Postfix configuration files can be found in **`/etc/postfix`** directory. The main configuration file is logically called **`main.cf`**

By default Postfix will only accept incoming emails from localhost. To accept emails from the complete network, configure the following option:

```
inet_interfaces = all
```

In order to make Postfix aware of the local emaildomains, add the following line to the **`main.cf`**.

```
virtual_mailbox_domains = example.com, example.org, example.net
```

Postfix will now see the configured domains as it's local email domains, however to accept incoming emails Postfix will do a recipient check. Add the following lines to the **`main.cf`** to have Postfix use LDAP for looking up (valid) recipients:

```
virtual_mailbox_maps = ldap:/etc/postfix/ldap-users.cf
virtual_alias_maps = ldap:/etc/postfix/ldap-aliases.cf
virtual_transport = lmtp:127.0.0.1:2003
```

All incoming emails are delivered to the LMTP service of the **`zarafa-dagent`**. The delivery needs to be done on the primary mail address of a user. For resolving the primary mail address of the user, create the file **`/etc/postfix/ldap-users.cf`** and add the following lines:

```
server_host = localhost
search_base = ou=Users,dc=example,dc=com
version = 3
scope = sub
query_filter = (&(objectClass=posixAccount)(mail=%s))
result_attribute = mail
```

For lookups of mail aliases create the file **`/etc/postfix/ldap-aliases.cf`** and add the following lines:

```
server_host = localhost
search_base = ou=Users,dc=example,dc=com
version = 3
scope = sub
query_filter = (&(objectClass=posixAccount)(zarafaAliases=%s))
result_attribute = mail
```

[2] http://www.postfix.org/LDAP_README.html

The search base of users and aliases need to match the search base of the LDAP server. After the configuration files have been changed Postfix need to be restarted:

```
/etc/init.d/postfix restart
```

Make sure the **zarafa-dagent** is run as a daemon and started at boot time.

For RPM based distributions use:

```
chkconfig zarafa-dagent on
/etc/init.d/zarafa-dagent start
```

For Debian based distributions enable the zarafa-dagent by setting the option DAGENT_ENABLED to **yes** in the file **/etc/default/zarafa-dagent**. To enable the **zarafa-dagent** at boot time use:

```
update-rc.d zarafa-dagent defaults
```

Note

It is advised to enable logging of the **zarafa-dagent** when running in LMTP mode for monitoring purposes. Enable the logging options in the **zarafa-dagent** in **/etc/zarafa/dagent.cfg**.

5.4.2. Configure ZCP Postfix integration with Active Directory

The Postfix can resolve primary mail addresses and aliases of users and groups from the Active Directory server. The Postfix package in most Linux distributions has LDAP support enabled by default. To read more about Postfix LDAP support see *the LDAP README*[3] on the Postfix website.

All Postfix configuration files can be found in **/etc/postfix** directory. The main configuration file is logically called **main.cf**.

By default Postfix will only accept incoming emails from localhost. To accept emails from the complete network, configure the following option:

```
inet_interfaces = all
```

In order to make Postfix aware of the local emaildomains, add the following line to the **main.cf**:

```
virtual_mailbox_domains = example.com, example.org, example.net
```

Postfix will now see the configured domains as it's local email domains, however to accept incoming emails Postfix will do a recipient check. This recipient check can be done on the Active Directory server. Add the following lines to the **main.cf**

```
virtual_mailbox_maps = ldap:/etc/postfix/ldap-users.cf
virtual_alias_maps = ldap:/etc/postfix/ldap-aliases.cf
virtual_transport = lmtp:127.0.0.1:2003
```

[3] http://www.postfix.org/LDAP_README.html

All incoming emails are delivered to the LMTP service of the **zarafa-dagent**. The delivery needs to be done on the primary mail address of a user. For resolving the primary mail address of the user, create the file **/etc/postfix/ldap-users.cf** and add the following lines:

```
server_host = 192.168.0.100
search_base = ou=Users,dc=example,dc=local
version = 3
bind = yes
bind_dn = cn=zarafa,ou=Users,dc=example,dc=local
bind_pw = secret
scope = sub
query_filter = (&(objectClass=user)(mail=%s))
result_attribute = mail
```

For lookups of mail aliases create the file **/etc/postfix/ldap-aliases.cf** and add the following lines:

```
server_host = 192.168.0.100
search_base = ou=Users,dc=example,dc=local
version = 3
bind = yes
bind_dn = cn=zarafa,ou=Users,dc=example,dc=local
bind_pw = secret
scope = sub
query_filter = (&(objectClass=user)(otherMailbox=%s))
result_attribute = mail
```

Active Directory has the possibility to create distribution groups which can be used as email distribution list in ZCP. To use integrate Postfix with distribution groups, Postfix 2.4 or higher is required.

Note

Some linux distributions (like RHEL 4 and 5) do not include Postfix 2.4 or higher. Packages of newer versions of Postfix are usually available as community contributed packages. In case of RHEL 4 and 5 these packages can be found *here*[4].

To support distribution groups add the following line to the **virtual_alias_maps**:

```
virtual_alias_maps = ldap:/etc/postfix/ldap-aliases.cf, ldap:/etc/postfix/ldap-groups.cf
```

Create a new file **/etc/postfix/ldap-group.cf** and insert the LDAP group configuration in there:

```
server_host = 192.168.0.100
search_base = ou=groups,dc=example,dc=local
version = 3
bind = yes
bind_dn = cn=zarafa,ou=Users,dc=example,dc=local
bind_pw = secret
query_filter = (&(objectclass=group)(mail=%s))
leaf_result_attribute = mail
special_result_attribute = member
```

[4] http://www.linuxmail.info/postfix-rpm-packages

The search base of users, aliases and groups need to match the search base of the Active Directory server. After the configuration files have been changed Postfix need to be restarted:

```
/etc/init.d/postfix restart
```

Make sure the **zarafa-dagent** is run as a daemon and started at boot time.

For RPM based distributions use:

```
chkconfig zarafa-dagent on
/etc/init.d/zarafa-dagent start
```

For Debian based distributions enable the zarafa-dagent by setting the option DAGENT_ENABLED to **yes** in the file **/etc/default/zarafa-dagent**. To enable the **zarafa-dagent** at boot time use:

```
update-rc.d zarafa-dagent defaults
```

Note

It is advised to enable logging of the **zarafa-dagent** when running in LMTP mode for monitoring purposes. Enable the logging options in the **zarafa-dagent** in **/etc/zarafa/dagent.cfg**.

5.4.3. Configure ZCP Postfix integration with virtual users

If no OpenLDAP or Active Directory Server is available, Postfix can be configured with virtual users in a hash map. In this section we explain how.

By default Postfix will only accept incoming emails from localhost. To accept emails from the complete network, configure the following option:

```
inet_interfaces = all
```

All Postfix configuration files can be found in **/etc/postfix** directory. The main configuration file is logically called **main.cf**

In order to make Postfix aware of the local email domains, add the following line to the **main.cf**:

```
virtual_mailbox_domains = example.com, example.org, example.net
```

Postfix will now regard these domains as it's local email domains. In order to accept incoming emails Postfix will also need to validate the recipient. Add the following lines to the **main.cf** config file in order to have Postfix look up recipient from a hash map:

```
virtual_mailbox_maps = hash:/etc/postfix/virtual
virtual_alias_maps = hash:/etc/postfix/virtual
virtual_transport = lmtp:127.0.0.1:2003
```

The file **/etc/postfix/virtual** should contain all email addresses and aliases of a user, in the following structure:

```
#Emailaddress or alias        primary mailaddress of user
john@example.com              john@example.com
```

```
user1@example.com             user1@example.com
user1@example.net             user1@example.com
alias_user1@example.com       user1@example.com
info@example.com              user2@example.com, user1@example.com
```

The left column contains the email address or alias, the right column contains the primary email addresses on which the message should be delivered.

After all users and aliases are added to this file, a hash map needs to be created. The following command will create the actual hash map **`/etc/postfix/virtual.db`**.

```
postmap /etc/postfix/virtual
```

All incoming emails are delivered to the **`zarafa-dagent`** over LMTP using the primary mail address of as specified in the hash map.

After changing the configuration files restart Postfix by its init script:

```
/etc/init.d/postfix restart
```

For RPM based distributions use:

```
chkconfig zarafa-dagent on
/etc/init.d/zarafa-dagent start
```

For Debian based distributions enable the zarafa-dagent by setting the option DAGENT_ENABLED to **`yes`** in the file **`/etc/default/zarafa-dagent`**. To enable the **`zarafa-dagent`** at boot time use:

```
update-rc.d zarafa-dagent defaults
```

Note

It's advised to enable logging of the **`zarafa-dagent`** when running in LMTP mode for monitoring purposes. To alter logging options for the **`zarafa-dagent`**, adjust the configuration file: **`/etc/zarafa/dagent.cfg`**.

5.5. Configure Z-Push (Remote ActiveSync for Mobile Devices)

This chapter describes how to configure the Z-Push software to bridge ZCP with ActiveSync enabled PDAs and smartphones.

Z-Push is available as an open source project on Sourceforge - *http://z-push.sourceforge.net*

In this manual only the server part of Z-Push is discussed, please refer to our User Manual for instruction on configuring mobile devices.

Mobile phones, smartphones and PDAs can be synchronized because Z-Push emulates the ActiveSync functionality of a MS Exchange server on the server side, allowing mobiles to synchronize via *over-the-air* ActiveSync (AirSync). Using Z-Push most mobiles can synchronize without installing any additional software on the device.

Z-Push needs to be installed on a web server. It is highly recommended to use Apache. It is also highly recommended to use PHP as an Apache module.

5.5.1. Compatibilité

Z-Push allows users with PDAs and smartphones to synchronise their email, contacts, calendar items and tasks directly from a compatible server over UMTS, GPRS, WiFi or other GSM data connections. Les appareils mobiles suivants sont pris en charge par Z-Push:

- Apple iPhone and iPad
- Windows Mobile 5, 6, 6.1 and 6.5
- Windows Phone 7 and 7.5
- Nokia E/N-series with Mail for Exchange (M4E)
- Nokia E-series with built in ActiveSync (Nokia Mail 2)
- Android Cupcake or Donut with third party tools like Nitrodesk Touchdown
- Android Eclair with Contacts and Calendar synchronization or third party tools
- Android Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich and Jelly Bean using the default ActiveSync client (Microsoft Exchange ActiveSync type account) or third party tools
- Blackberry PlayBook
- other ActiveSync compatible devices

For detailed information about the devices and their compatibility status, please consult the Mobile Compatibility List at *http://z-push.sourceforge.net/compatibility*

5.5.2. Sécurité

To encrypt data between the mobile devices and the server, it's required to enable SSL support in the web server. Configuring Apache with SSL certificates is beyond the scope of this document, though many howtos can be found online.

Keep in mind that some mobile devices require an official SSL certificate and don't work with self signed certificates. For Windows Phone and Windows Mobile you might need to install the certificates on the device (See *Section 5.6, « Configuring SSL for Windows Mobile and Windows Phone »* for details).

5.5.3. Installation

Download the latest Z-Push software from *http://z-push.sourceforge.net/download*

To install Z-Push, simply untar the Z-Push tar to the webroot with:

```
tar zxvf z-push-<version>.tar.gz -C /var/www/html
```

The **-C** option is the destination where the files need to be installed. In the following table the default webroot directories of where some distributions lets the Apache webserver search for files.

Tableau 5.2. Webroot directories

Distribution	Répertoire racine Web par défaut
Red Hat Enterprise Linux	/var/www/html
SLES	/srv/www/htdocs

Distribution	Répertoire racine Web par défaut
Debian et Ubuntu	/var/www

Make sure that the ‘state’ directory exists and is writeable for the webserver process, so either change the owner of the ‘state’ directory to the UID of the apache process, or make it world writeable:

For Z-Push 2.X: chmod 755 /var/lib/z-push/ chown apache:apache /var/lib/z-push/

For Z-Push 1.X chmod 755 /var/www/z-push/state chown apache:apache /var/www/z-push/state

Le nom d'utilisateur ou de groupe de Apache peut différer d'une distribution Linux à l'autre. The table below shows an overview of the user and group names of the Apache process.

Tableau 5.3. User and groupnames per distribution

Distribution	Nom d'utilisateur Apache	Nom de groupe
Red Hat Enterprise Linux	apache	apache
SLES	wwwrun	www
Debian et Ubuntu	www-data	www-data

On systems with SELinux you might need to change the security context of this folder, e.g. chcon -R -t httpd_sys_rw_content_t /var/lib/z-push

Now, Apache must be configured to redirect the URL **`Microsoft-Server-ActiveSync`** to the **`index.php`** file in the z-push directory. This can be done by adding the line to the **`httpd.conf`** file

```
Alias /Microsoft-Server-ActiveSync /var/www/html/z-push/index.php
```

Veuillez vous assurer que cette ligne soit ajoutée dans la section appropriée de votre configuration Apache, celle comportant les 'virtual hosts' et autres configurations Apache.

Important

It is not possible simply rename the **`Z-Push`** directory to **`Microsoft-Server-ActiveSync`**. This will cause Apache to send redirects to the smartphone, which will definitely prevent proper synchronization.

Enfin, assurez vous que PHP comporte les paramètres suivants:

```
php_flag magic_quotes_gpc = off
php_flag register_globals = off
php_flag magic_quotes_runtime = off
php_flag short_open_tag = on
```

Set this in the **`php.ini`** or in a **`.htaccess`** file in the root directory of Z-Push. If not setup correctly, the smartphone will not be able to login correctly via Z-Push.

Reload Apache to activate these changes.

In Z-Push 2.X versions the default log directory is **`/var/log/z-push`**. Make sure this directory exists and is writeable for the webserver process. On systems with SELinux you might need to change the security context of this folder, e.g. chcon -R -t httpd_sys_rw_content_t /var/log/z-push

5.5.4. Mobile Device Management

Users can remote wipe own mobile devices from the ZCP Webaccess without interaction of the system administrator. The Mobile Device Management (MDM) plugin can be downloaded at: *http://www.zarafa.com/integrations/mobile-device-management-plugin*

The system administrator can remote wipe devices from the command line using the **`z-push-admin`** tool.

5.5.5. Mise à niveau

Upgrading to a newer Z-Push version follows the same path as the initial installation.

When upgrading to a new minor version e.g. from Z-Push 1.4 to Z-Push 1.4.1, the existing Z-Push directory can be overwritten when extracting the archive. When installing a new major version it is recommended to extract the tarball to another directory and to copy the state from the existing installation.

Important

It is crucial to always keep the data of the state directory in order to ensure data consistency on already synchronized mobiles.

Without the state information mobile devices, which already have an ActiveSync profile, will receive duplicate items or the synchronization will break completely.

Important

Upgrading to Z-Push 2.X from 1.X it is not necessary to copy the state directory because states are not compatible. However Z-Push 2 implements a fully automatic resynchronizing of devices in the case states are missing or faulty.

Important

Downgrading from Z-Push 2.X to 1.X is not simple. As the states are not compatible you would have to follow the procedure for a new installation and re-create profiles on every device.

Please also observe the published release notes of the new Z-Push version. For some releases it is necessary to e.g. resynchronize the mobile.

5.6. Configuring SSL for Windows Mobile and Windows Phone

If you don't have a certificate of one of the Certified Authorities, you also need to add the CA Certificate to the Trusted Root Certificates store of the device.

The certificates should be in DER format to install it on a windows device. By default the generated SSL certificates on Linux are in PEM format. The DER certificate is a base64 encoded PEM certificate. You can convert the certificate type by the following commands: openssl x509 -in ca.crt -inform PEM -out ca.cer -outform DER openssl x509 -in host.crt -inform PEM -out host.cer -outform DER

where **ca.crt** is your CA certificate file and **host.crt** is your certified file.

After converting both certificates you need to copy them to the PDA. It can be e.g. done by putting the files on a local intranet server and accessing them with the device's browser: *http://intranet/certs/ca.cer http://intranet/certs/host.cer* By selecting the certificates on your PDA they will be stored in the Trusted Root Certificates store of your device.

5.6.1. Résolution d'erreurs

General configuration

Most of the difficulties are caused by incorrect Apache settings. The Apache setup can be tested using a webbrowser like Firefox pointing it to:

```
http://<server>/Microsoft-Server-ActiveSync
```

If correctly configured, a window requesting username/password should be displayed. Authenticating using valid credentials should display Z-Push information page, containing the following message:

A Z-Push information page should be displayed, containing the message:

```
*GET not supported*
This is the z-push location and can only be accessed by Microsoft ActiveSync-capable devices.
```

Verify the PHP and/or Apache configuration if an error is displayed.

Synchronization problems

Important

The following text regarding debug.txt and WBXML debug applies to Z-Push 1.X versions only. In Z-Push 2 there is a separate log directory and the loglevel is configured in **config.php**.

If synchronization problems are encountered, a **debug.txt** file has to be created in the root directory of Z-Push. This file should be writeable by the Apache server process.

```
touch /var/www/z-push/debug.txt
chmod 777 /var/www/z-push/debug.txt
```

The **debug.txt** file will collect debug information about the synchronisation.

To obtain a complete synchronization log the file **wbxml.php** has to be edited and the parameter **WBXML_DEBUG** set to true:

```
define('WBXML_DEBUG', true);
```

Important

The **debug.txt** logfile **contains sensitive data and should be protected** so it can not be downloaded from the internet.

To protect the **`debug.txt`** logfile, a **`.htaccess`** has to be created in the z-push root directory, containing:

```
<Files debug.txt>
 Deny from All
</Files>
```

Log messages

- **Repeatedly “Command denied: Retry after sending a PROVISIONING command”:**

Most probably the mobile device does not support provisioning. The LOOSE_PROVISIONING parameter should be enabled in the configuration. If the messages continues, the ActiveSync profile should be reconfigured on the device. If this does not help, the PROVISIONING could be disabled completely in the config file (applies to all devices!). More information can be found at: *http://www.zarafa.com/wiki/index.php/Z-Push_Provisioning*

- **Exceptions for Meeting requests cause duplicates if accepted on the mobile:**

Please update to Z-Push 1.4 or later. In order to fix existing duplicates, the ActiveSync profile on the mobile has to be recreated or at least the calendar has to be resynchronized completely (disabling calendarsync and enabling it afterwards).

Configurations avancées

Ce chapitre décrit la configuration d'environnements particuliers qui sortent du cadre ordinaire des installations classiques de ZCP.

6.1. Exécution des composants ZCP en dehors de l'hôte local

Lors d'une connexion SSL avec certificats, il sera non seulement possible de crypter les connexions, mais les services Linux pourront également s'identifier de manière sécurisée à l'aide du certificat du client SSL.

Répéter l'opération de création de certificat pour chacun des programmes clients tels que **`zarafa-spooler`**, **`zarafa-monitor`**, **`zarafa-gateway`**, **`zarafa-dagent`** et **`zarafa-admin`**. Il est possible de créer un certificat global pour tous ces programmes, ou bien de créer un certificat distinct pour chacun d'eux séparément. Ces clients pourront alors effectuer des connexions SSL en utilisant leur certificat comme méthode d'authentification.

```
sh /usr/share/doc/zarafa/ssl-certificates.sh client
```

Lors du renseignement des informations du certificat, il faut au minimum s'assurer que le champ Organizational Unit Name soit différent de celui des autres certificats. Il ne faut pas oublier non plus de renseigner le champ Common Name.

Lors de la demande de confirmation de création de clé publique, saisir y ou o (pour yes/oui) et appuyer sur Entrée. Un nouveau certificat nommé **`client.pem`** et une clé publique nommée **`client-public.pem`** sont maintenant présents. Comme exemple, les options de configuration devant être modifiées dans le fichier **`dagent.cfg`** sont les suivantes :

```
server_socket = https://name-or-ip-address:237/zarafa
sslkey_file = /etc/zarafa/ssl/client.pem
sslkey_pass = ssl-client-password
```

Important

Pour que l'utilitaire **`zarafa-admin`** puisse fonctionner correctement dans une architecture multi-serveur, le fichier **`admin.cfg`** doit impérativement être situé dans le répertoire de configuration de ZCP, généralement **`/etc/zarafa/`**. Ce fichier doit contenir les options citées ci-dessus.

Saisir le nom d'hôte ou l'adresse IP adéquate pour l'option server_socket. Si un numéro de port différent est utilisé pour les connexions SSL du serveur, saisir également le numéro de port adéquat. Remplacer le mot de passe par celui qui a été déterminé lors de la création du certificat.

Copier le fichier **`client-public.pem`** vers l'emplacement du serveur :

```
mkdir /etc/zarafa/sslkeys
mv client-public.pem /etc/zarafa/sslkeys
```

Désormais, le client connaît la clé privée et le serveur connaît la clé publique. Le client peut se connecter au serveur à l'aide de cette clé à partir de n'importe quel emplacement du réseau ou d'Internet.

Note

Il faut porter une attention particulière au fichier **`client.pem`**. Quiconque ayant cette clé privée pourra se connecter au serveur Zarafa et en deviendra l'utilisateur interne SYSTEM, pouvant faire ce qu'il souhaite sans aucune restriction.

6.2. Configurations multi-tenant

Cette section procure des informations concernant la fonctionnalité multi-tenant apportée par la version 6.10 de Zarafa. Cette fonctionnalité est disponible dans toutes les éditions, mais la prise en charge officielle n'est uniquement assurée que dans l'édition 'Entreprise' et dans les éditions hébergées .

Le mode multi-tenant permet de gérer plusieurs entreprises différentes sur un seul serveur ZCP sans que les membres d'une entreprise ne puisse voir ceux d'une autre entreprise.

6.2.1. Prise en charge des plugins de gestion des utilisateurs

La prise en charge multi-tenant ne peut être activée que si le plugin DB ou LDAP est utilisé. Pour le moment, il n'est pas possible d'utiliser le plugin Unix. Avec l'utilisation du plugin DB, la gestion des tenants (sociétés) pourra s'effectuer à l'aide de l'utilitaire **`zarafa-admin`**, tandis qu'avec l'utilisation du plugin LDAP toutes les informations proviendront directement de LDAP ou Active Directory.

Il est préférable d'avoir recours au plugin LDAP pour la gestion des environnements multi-tenant.

6.2.2. Configuration du serveur

Les options suivantes de configuration du fichier **`server.cfg`** seront utilisées lors de l'activation d'un environnement multi-tenant.

```
enable_hosted_zarafa
```

La valeur **`true`** permettra de créer plusieurs tenants au sein de l'instance Zarafa et d'assigner tous les utilisateurs et groupes à des tenants spécifiés. La valeur **`false`** créera l'environnement habituel utilisé par un tenant unique.

```
createcompany_script
```

L'emplacement du script **`createcompany`** qui sera exécuté lorsqu'une nouveau tenant sera créé.

```
deletecompany_script
```

L'emplacement du script **`deletecompany`** qui sera exécuté lorsqu'une nouveau tenant sera supprimé.

```
loginname_format
```

Consulter *Section 6.2.2.2, « Configuration de l'identifiant de connexion »* pour plus de détails sur cette option de configuration.

```
storename_format
```

Consulter *Section 6.2.2.3, « Configuration du nom de la base de stockage »* pour plus de détails sur cette option de configuration.

6.2.2.1. Activation du mode multi-tenant

Pour activer le mode multi-tenant dans Zarafa, modifier l'option suivante dans le fichier **server.cfg** :

```
enable_hosted_zarafa = yes
```

6.2.2.2. Configuration de l'identifiant de connexion

L'identifiant de connexion d'un utilisateur doit être unique afin de permettre la tentative de connexion. Lors de l'activation du mode multi-tenant dans Zarafa, l'unicité des identifiants de connexion peut devenir difficile si le nombre des entités augmente. Une solution simple consiste à incorporer les *noms d'entité* dans les *identifiants de connexion* afin de forcer l'unicité des *identifiants de connexion*.

La méthode avec laquelle le *nom d'une société* peut être 'attaché' au nom d'un utilisateur afin de créer un identifiant de connexion peut être configurée à l'aide de l'option **loginname_format** du fichier **server.cfg**. Cette option de configuration peut contenir les variables suivantes :

- **%u** - Le *nom d'utilisateur*
- **%c** - Le *nom de la société* à laquelle l'utilisateur appartient

Comme caractère de séparation entre le *nom d'utilisateur* et le *nom de l'entité*, un symbole doit être choisi qui ne figure ni dans le *nom d'utilisateur* ni dans le *nom de l'entité*. Par exemple, les caractères **@** et **** sont valides.

Voici quelques exemples de formatage d'un *identifiant de connexion* pour un utilisateur nommé "John Doe" qui est membre de "Exampleorg":

- **%u >** john
- **%u@%c >** *john@exampleorg*[1]
- **\\%c\%u >** \\exampleorg\john

Bien qu'il soit obligatoire d'utiliser un *identifiant de connexion* contenant un **%c** lorsque le plugin DB est utilisé, c'est facultatif si le plugin LDAP est utilisé. La gestion des *identifiants de connexion uniques est plus aisée dans LDAP parce qu'il est possible d'utiliser une adresse de courrier électronique en tant qu'attribut de l'identifiant de connexion*. Consulter le fichier de configuration LDAP plus d'information sur l'attribut **loginname**.

Note

Lorsqu'un nom d'utilisateur est utilisé comme paramètre avec l'utilitaire **zarafa-admin** il doit présenter le même formatage que celui qui a été défini dans la configuration. Par exemple si la valeur de la configuration de l'option **loginname_format** comporte la variable du nom de l'organisme (**%c**), alors le nom de l'organisme devra également être présent chaque fois qu'un nom d'utilisateur est utilisé avec l'utilitaire **zarafa-admin**.

6.2.2.3. Configuration du nom de la base de stockage

Si les communications sont autorisées entre différents tenants (sociétés), il est alors possible pour les utilisateurs d'un tenant de partager leur base avec les utilisateurs d'un autre tenant. Pour différencier

plus facilement les bases appartenant à des tenants différents, le nom d'une base de stockage peut être formaté afin de contenir le nom du tenant (*nom de l'organisme*) auquel l'utilisateur ou la base appartient.

Dans le fichier **`server.cfg`** l'option de configuration **`storename_format`** est prévue à cet effet. Ce format permet l'utilisation d'une série de variables correspondant à diverses informations :

- **`%u`** — Le *nom d'utilisateur*
- **`%f`** — Les *nom et prénom* de l'utilisateur
- **`%c`** — Le *nom de l'organisme*, nom du tenant, auquel l'utilisateur appartient

Voici quelques exemples pour un utilisateur nommé ‘John Doe’ qui est membre du tenant ‘Exampleorg’ :

- **`%u >`** john
- **`%f >`** John Doe
- **`%f (%c) >`** John Doe (Exampleorg)

6.2.2.4. Configuration du plugin LDAP

Si vous utilisez le plugin DB, aucune configuration supplémentaire n'est nécessaire. Pour le plugin LDAP, certaines options de configuration nécessiteront sans doute une modification.

Pour un environnement LDAP multi-tenant, il est nécessaire d'avoir les différentes sociétés dans l'arborescence LDAP ainsi que, sous chaque container de société, les utilisateurs, groupes et contacts de la société en question. Il impossible d'assigner une utilisateur à une société spécifique par à l'aide d'un attribut LDAP.

Consulter la capture d'écran ci-dessous pour un exemple de structure LDAP.

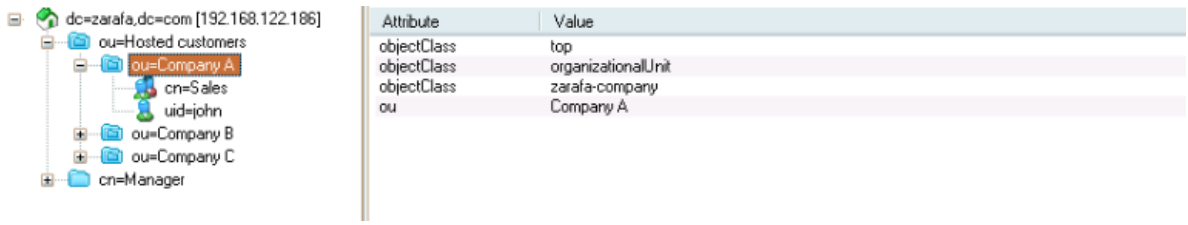


Figure 6.1. Arborescence LDAP d'un environnement multi-tenant

Modifier les lignes suivantes dans le fichier de configuration LDAP, afin de paramétrer le mode multi-tenant.

```
ldap_company_unique_attribute = ou
ldap_companyname_attribute = ou
ldap_company_scope = sub
```

Tester les paramètres à l'aide de **`zarafa-admin --list-companies`** et **`zarafa-admin -l`**.

S'il n'y a aucune société ni utilisateur, veuillez rechercher les erreurs dans l'historique du fichier journal du serveur Zarafa. Définir le niveau de journalisation sur **`6`** dans le fichier de configuration **`/etc/zarafa/server.cfg`** affichera l'historique de toutes les requêtes LDAP et de toutes les erreurs possibles.

Lorsque le mode multi-tenant est activé, il est non seulement possible d'avoir différents organismes sur le même serveur, mais certains paramètres plus avancés peuvent également être configurés, tels que les délégations de courrier inter-organismes, les différentes hiérarchies d'administrateur et les niveaux des quotas par organisme.

Consulter la page man de **`zarafa-ldap.cfg`** pour plus d'information à propos des fonctionnalités LDAP du mode multi-tenant.

```
man zarafa-ldap.cfg
```

6.2.2.5. Bases de stockage publiques

Une fois que le serveur a été démarré correctement, les bases peuvent être crées. Il y a deux types de bases: les bases privées et les bases publiques. Il ne peut y avoir qu'une seule base de stockage publique pour chaque société. Lorsque vous créer une société, sa base publique sera simultanément créée. Si pour une raison quelconque, la base publique d'une société donnée n'était pas créée automatiquement, il sera possible de le faire manuellement à l'aide de la commande suivante :

```
/usr/bin/zarafa-admin -s -I <tenant>
```

Remplacer **`<tenant>`** avec le nom du tenant (la société) pour laquelle la base publique doit être créée. Si l'option **`-I`** n'est pas utilisée, la base publique sera créée pour un environnement de tenant unique (et ne sera pas accessible lorsque l'environnement multi-tenant de Zarafa sera activé). Par défaut, le dossier public d'un tenant est accessible à tous les utilisateurs de ce tenant (société).

6.2.3. Gestion des tenants (sociétés)

Note

La gestion des tenants (sociétés) à l'aide de**`zarafa-admin`** n'est disponible que si le plugin DB est utilisé. Si le plugin LDAP est utilisé, toutes les opérations d'administration peuvent être effectuées à l'aide du serveur Active Directory ou LDAP.

Pour créer une nouvelle société, exécuter la commande suivante :

```
/usr/bin/zarafa-admin --create-company <companyname>
```

Pour supprimer une société, exécuter la commande suivante :

```
/usr/bin/zarafa-admin --delete-company <companyname>
```

Pour modifier une société, exécuter la commande suivante :

```
/usr/bin/zarafa-admin --set-company <companyname>
```

Cette commande peut être couplée avec l'option **`--qw`** pour définir le seuil d'alerte de dépassement de quota pour une société donnée.

Pour administrer les permissions d'accès d'une société, les commandes suivantes peuvent être utilisées :

```
/usr/bin/zarafa-admin --add-view <viewer> -I <companyname>
/usr/bin/zarafa-admin --del-view <viewer> -I <companyname>
/usr/bin/zarafa-admin --list-view -I <companyname>
```

L'intitulé **<viewer>** doit être remplacé par le nom de la société qui reçoit ou qui perd la permission de 'voir' la société **<companyname>**. À l'aide des privilèges d'accès, le carnet d'adresses global peut être partagé entre plusieurs organisations et des délégations de boîte aux lettres inter-organisation peuvent être définies.

```
/usr/bin/zarafa-admin --add-admin <admin> -I <companyname>
/usr/bin/zarafa-admin --del-admin <admin> -I <companyname>
/usr/bin/zarafa-admin --list-view -I <companyname>
```

L'intitulé**<admin>** doit être remplacé par l'identifiant de connexion de l'utilisateur qui reçoit ou qui perd les privilèges d'administration sur la société **<companyname>**.

6.2.4. Gestion des utilisateurs et des groupes

Si le plugin DB est utilisé, les utilisateurs et les groupes devront être créés à l'aide de l'utilitaire **zarafa-admin**. Pour plus d'information sur l'utilisation de l'utilitaire **zarafa-admin**, veuillez exécuter **man zarafa-admin**. Le format du nom de l'utilisateur ou du groupe qui sera donné par l'utilitaire **zarafa-admin** dépend de l'option de configuration **loginname_format**.

Par exemple, si l'option **loginname_format** était définie sur **%u@%c** la création d'un utilisateur pour le tenant **exampleorg** s'effectuerait de la façon suivante :

```
/usr/bin/zarafa-admin --c john@exampleorg ...autres options...
```

Et la création d'un nouveau groupe pour le tenant **exampleorg** s'effectuerait ainsi :

```
/usr/bin/zarafa-admin -g group@exampleorg ...autres options...
```

6.2.5. Niveaux de quota

Lors de l'utilisation d'une installation multi-tenant, il y a 2 types de quotas, à savoir le quota s'appliquant au tenant (la société) et le quota s'appliquant à l'utilisateur individuel. Le quota s'appliquant au tenant est vérifié sur la totalité du stockage de tous les utilisateurs au sein de ce tenant en plus de la base publique de stockage.

Actuellement, seul le quota d'avertissement est applicable à un tenant, il est donc impossible de définir un quota modéré ou strict afin de limiter les capacités de stockage des courriers d'un tenant.

De même que pour les quotas d'utilisateur, il existe plusieurs niveaux de quotas de tenants, et il y existe même un nouveau niveau de quota d'utilisateur. Voici le sommaire des niveaux de quotas pouvant être définis dans un environnement multi-tenant :

1. Quota tenant (société) :
    a. **Global company quota** : Configuré dans **/etc/zarafa/server.cfg** et affecte tous les tenants du système.
    b. **Specific company quota** : Le niveau de quota pour d'un tenant, configuré à l'aide du plugin LDAP (ou de l'utilitaire **zarafa-admin**).

2. Quota utilisateur :

   a. **`Global user quota`** : Configuré dans **`/etc/zarafa/server.cfg`** et affecte tous les utilisateurs de tous les tenants.

   b. **`Company user quota`** : C'est le niveau de quota par défaut pour tous les utilisateurs d'un tenant donné, il est configuré à l'aide du plugin de gestion des utilisateurs au niveau du tenant.

   c. **`Specific user quota`** : C'est le niveau de quota pour un utilisateur donné, il est configuré à l'aide du plugin de gestion des utilisateurs.

Comme indiqué ci-dessus, les paramétrages **`Global company quota`** et **`Global user quota`** peuvent être configurés dans le fichier **`/etc/zarafa/server.cfg`** à l'aide des options **`quota_warn`**, **`quota_soft`** et **`quota_hard`** pour le quota d'utilisateur et de l'option **`companyquota_warn`** pour le quota de tenant.

Il est possible de configurer **`Specific company quota`** à l'aide de l'utilitaire **`zarafa-admin`** si le plugin DB est utilisé. La commande suivante définira les différents niveaux de quota affectant un tenant :

```
zarafa-admin --update-company <tenant> --qo y --qw <warningquota>
```

Il est possible de configurer **`Specific user quota`** à l'aide de l'utilitaire **`zarafa-admin`** si le plugin DB est utilisé. La commande suivante définira les différents niveaux de quota affectant un utilisateur :

```
zarafa-admin -u <user> --qo y --qh <hardquota> --qs <softquota> --qw <warningquota>
```

Il est possible de configurer **`Company user quota`** à l'aide de l'utilitaire **`zarafa-admin`**, si le plugin DB est employé, en utilisant l'argument **`--update-company`**. La commande suivante définira les différents niveaux de quota d'utilisateurs affectant un tenant :

```
zarafa-admin --update-company <tenant> --udqo y --udqh <hardquota> --udqs <softquota> --udqw
 <warningquota>
```

Si vous utilisez le plugin LDAP, les attributs contrôlant les niveaux de quota peuvent être configurés dans le fichier **`/etc/zarafa/ldap.cfg`**.

6.2.6. Administrateurs

Dans une installation multi-tenant, il y a deux sortes d'administrateurs :

- L'administrateur du système entier
- L'administrateur d'une des entreprises

L'administrateur système peut accéder à toutes les boîtes aux lettres du système entier. L'administrateur d'une entreprise ne peut accéder qu'aux boîtes aux lettres de sa propre entité.

Un administrateur système peut être créé en définissant la valeur de l'attribut **`zarafaAdmin`** d'un utilisateur sur 2 lorsque LDAP est utilisé, ou sur -a 2 lorsque le plugin DB est utilisé. Un administrateur entreprise peut être créé en définissant la valeur de l'attribut **`zarafaAdmin`** d'un utilisateur sur 1.

Le type d'administration attribué à un utilisateur peut être affiché à l'aide de l'utilitaire zarafa-admin :

```
zarafa-admin --details <admin username>
```

```
Username:               admin@example.com
Fullname:               Administrator
Emailaddress:           admin@example.com
Active:                 yes
Administrator:          yes (system)
```

6.3. Configuration multi-serveur

Ce chapitre présente la fonctionnalité multi-serveur qui fut apportée par la version 6.30 de Zarafa.

Note

Pour pouvoir utiliser cette fonctionnalité, une clé de licence valide Zarafa Entreprise est nécessaire, et le service zarafa-licensed doit obligatoirement être en exécution.

6.3.1. Introduction

La fonctionnalité multi-serveur ZCP permet de déployer ZCP sur de multiples serveurs. Dans ce cas, les bases de stockage des utilisateurs Zarafa sont divisées entre plusieurs serveurs, mais se comportent néanmoins comme dans un seul système centralisé. Les utilisateurs, les groupes et les tenants (sociétés) doivent être gérés dans un serveur LDAP ou Active Directory.

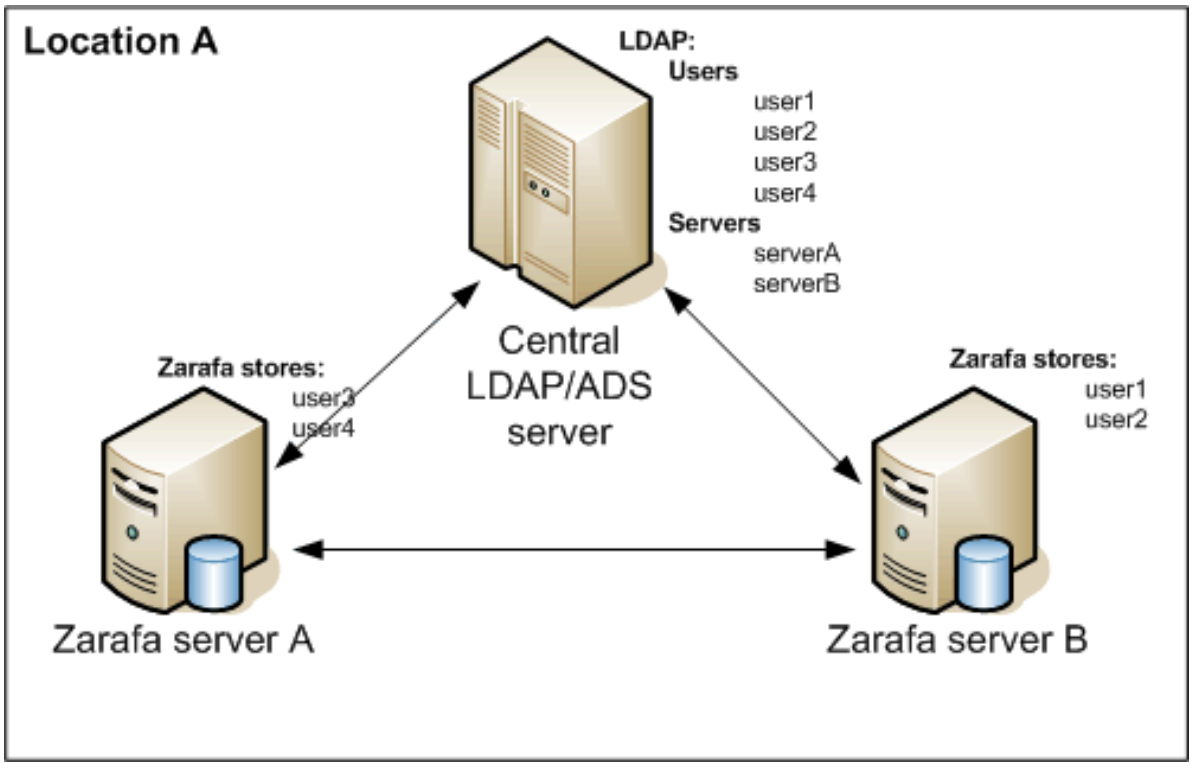


Figure 6.2. Environnement multi-serveur sur un emplacement unique

Le mode multi-serveur peut également être utilisé pour gérer un grand nombre d'utilisateurs ou pour distribuer les boîtes aux lettres sur plusieurs emplacements géographiques. Consulter *Figure 6.3, « Environnement multi-serveur déployé sur deux emplacements »*.

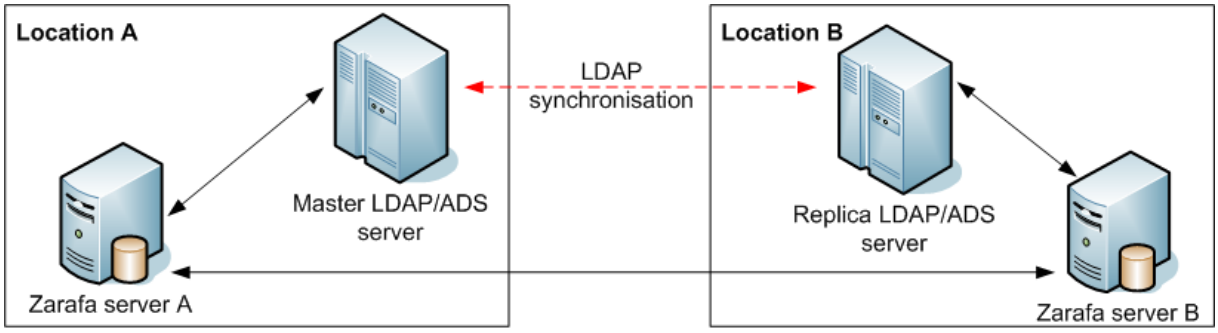


Figure 6.3. Environnement multi-serveur déployé sur deux emplacements

La boîte aux lettres d'un utilisateur est toujours stockée sur un seul serveur. Il est impossible de synchroniser des boîtes aux lettres déployées sur différents serveurs.

Lors d'un accès à de multiples boîtes aux lettres stockées sur plusieurs serveurs différents, le client établira une connexion avec les différents nœuds multi-serveur. Consulter le diagramme *Figure 6.4, « Environnement multi-serveur »*.

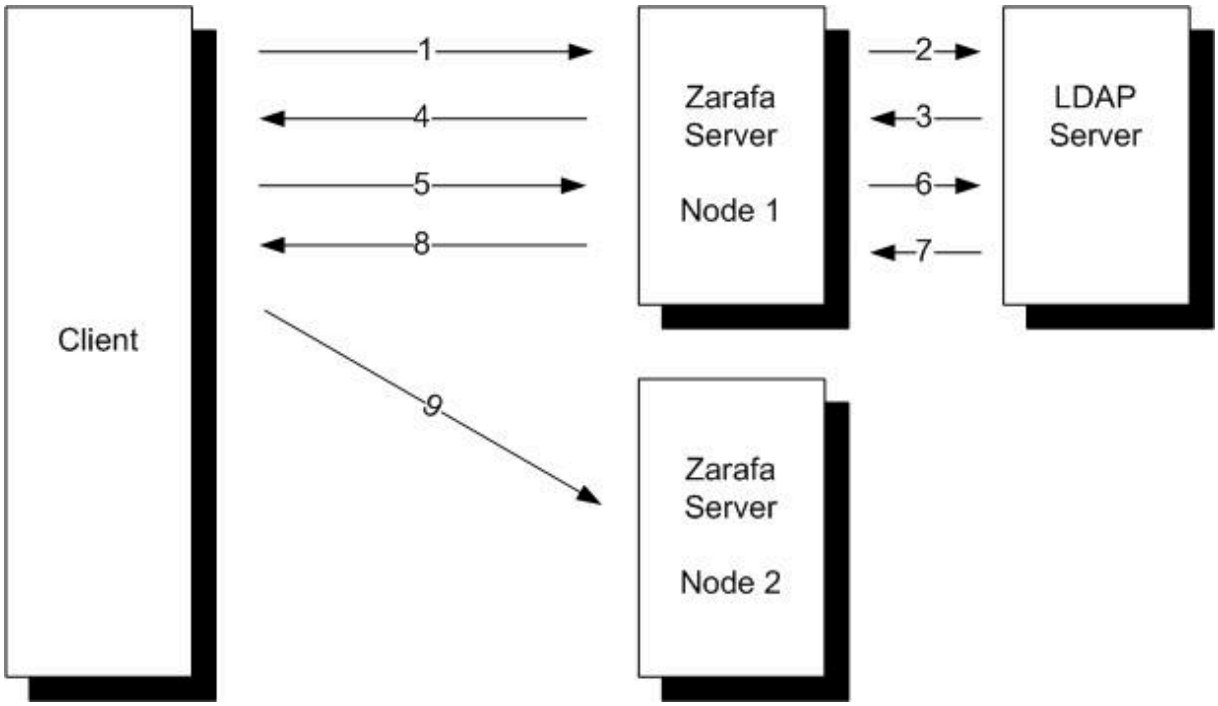


Figure 6.4. Environnement multi-serveur

L'utilisateur *John* est situé sur le *Nœud 1* et l'utilisatrice *Mary* est située sur le *Nœud 2*. John possède les droits d'accès en lecture sur la boîte aux lettres de Mary.

1. *John* démarre son client Outlook, qui se connecte sur le *Nœud 1*.
2. Le *Nœud 1* du serveur Zarfa vérifie l'attribut du serveur personnel dans le serveur LDAP central.
3. Les coordonnées du serveur hébergeant l'utilisateur *John* sont envoyées au serveur Zarafa.
4. La boîte aux lettres de *John's* est située sur le *Nœud 1*, par conséquent, elle sera chargée directement.
5. *John* demande au serveur Zarafa d'ouvrir la boîte aux lettres de *Mary*.
6. Le *Nœud 1* du serveur Zarafa vérifie l'attribut du serveur personnel de *Mary* dans le serveur LDAP central.

7. Les coordonnées du serveur hébergeant *Mary* sont envoyées au serveur Zarafa
8. Une demande de redirection est renvoyée au client
9. Le client établit une connexion avec le *Nœud 2* afin d'ouvrir la boîte aux lettres de *Mary*.

In the above example the client has a connection open to both nodes to access the mailboxes.

6.3.2. Préparation / configuration du serveur LDAP dans un environnement multi-serveur

La version multi-serveur de Zarafa ne peut être utilisée qu'avec le plugin de gestion des utilisateurs LDAP.

Dans un environnement multi-serveur, le serveur Zarafa interrogera le serveur LDAP non-seulement sur les utilisateurs et les groupes, mais également sur les nœuds de réseau.

1. Configuration du serveur LDAP à l'aide de *Section 5.2, « Configuration de l'intégration ZCP OpenLDAP »* ou *Section 5.3, « Configuration de l'intégration ZCP Active Directory »* dans ce manuel.
2. Ajouter à la structure LDAP un dossier ou une Organization Unit pour les différents nœuds du réseau multi-serveur Zarafa.

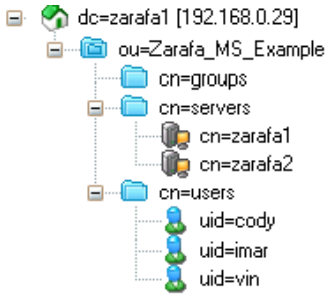


Figure 6.5. Configurer le répertoire avec tous les nœuds du réseau multi-serveur

3. Add all the multi-server nodes to this directory or organization unit. In Active Directory the **`Computer`** template can be used for this. When using OpenLDAP a custom LDAP object can be created, with the **`device`**, **`ipHost`** and **`zarafa-server`** *objectClass*.

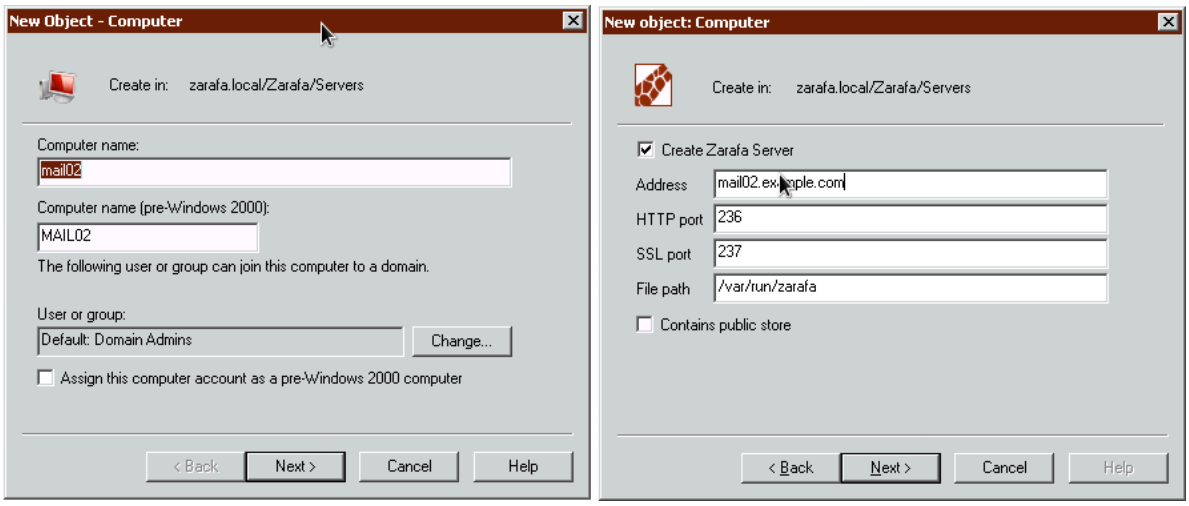


Figure 6.6. Computer creation wizard in ADS

4. Chaque nœud du réseau multi-serveur doit avoir un **`common name`**, **`FQDN`** ou **`ip-address`** et les **`coordonnées du serveur Zarafa`**. S'assurer que le **`FQDN`** puisse toujours être résolu par les clients.

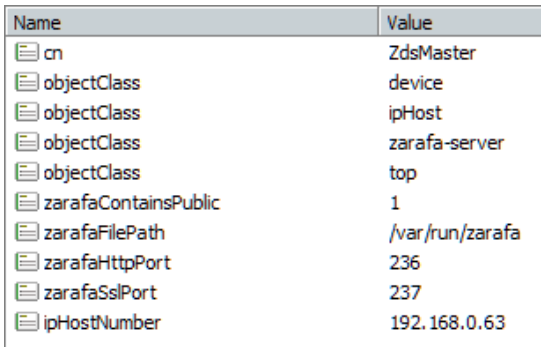


Figure 6.7. Attributs du serveur LDAP

5. L'attribut **`ZarafaContainsPublic`** peut uniquement être configuré pour un seul nœud du réseau multi-serveur. Pour le moment il n'est pas possible de déployer de multiples dossiers publics sur différents nœuds du réseau.

6. The Zarafa LDAP configuration needs to be extended with some extra multi-server configuration options. An example configuration file for the multi-server setup can be found in the **`/usr/share/doc/zarafa-multiserver/example-config`** directory. The files **`ldapms.*.cfg`** are the specific multi-server configuration files. The following LDAP configuration entries need to be configured for a multi-server setup:

```
ldap_server_type_attribute_value = zarafa-server
ldap_user_server_attribute = zarafaUserServer
ldap_server_address_attribute = ipHostNumber
ldap_server_http_port_attribute = zarafaHttpPort
ldap_server_ssl_port_attribute = zarafaSslPort
ldap_server_file_path_attribute = zarafaFilePath
ldap_server_search_filter =
ldap_server_unique_attribute = cn
```

7. Chaque utilisateur créé dans le serveur LDAP doit être assigné à l'un des nœuds du serveur Zarafa. Ceci peut être effectué à l'aide de l'attribut ZarafaUserServer. L'attribut doit contenir le nom unique du serveur.

Dans une situation multi-tenant, tous les tenants (sociétés) créés à l'aide de LDAP doivent être actualisés avec l'attribut **`zarafaCompanyServer`**. Utiliser également le nom du serveur pour cela.

6.3.3. Configuration des serveurs

Les options de configuration suivantes dans **`server.cfg`** sont fournies pour la prise en charge multi-serveur.

```
enable_distributed_zarafa
```

Activer l'environnement multi-serveur. Avec la valeur **`true`** il est possible de déployer des utilisateurs et des sociétés sur de multiples serveurs. Avec la valeur **`false`**, l'environnement pour une société unique est créé.

```
server_name
```

Le nom unique du serveur utilisé pour identifier chaque nœud du système. Ce nom de serveur doit être correctement configuré dans votre DNS. Ce nom de serveur doit être identique à la valeur de l'attribut **`zarafaUserServer`**.

Pour activer le mode multi-serveur dans Zarafa, modifier les options suivantes dans le fichier de configuration **`server.cfg`** :

```
user_plugin = ldapms
enable_distributed_zarafa = yes
server_name = <servername>
server_ssl_enabled = yes
```

Note

La migration d'un serveur unique vers un environnement multi-serveur n'est pas une simple formalité. Veuillez vérifier avec le support technique de Zarafa si la migration est possible pour le système utilisé.

6.3.4. Création de certificats SSL

Dans un environnement multi-serveur, il est indispensable de configurer la gestion SSL parce que des clients comme **`zarafa-dagent`**, **`zarafa-admin`** et **`zarafa-monitor`** doivent avoir un certificat SSL afin de pouvoir se connecter aux différents nœuds du réseau multi-serveur.

Il est tout d'abord nécessaire de créer les certificats du coté des serveurs, afin que le serveur Zarafa soit capable d'accepter des connexions SSL. Pour l'authentification des clients Linux tels que **`zarafa-dagent`**, une paire de clés, une publique et une privée, doit être créée.

Suivre les étapes ci-dessous afin de créer les certificats des serveurs ainsi que des clients.

1. Premièrement, créer le répertoire qui accueillera les certificats.

   ```
   mkdir /etc/zarafa/ssl
   chmod 700 /etc/zarafa/ssl
   ```

2. Créer le certificat du serveur à l'aide du script **`ssl-certificates.sh`** situé dans le répertoire **`/usr/share/doc/zarafa`**, qui utilise la commande openssl ainsi que le script **`CA.pl`**. Avant qu'un certificat de serveur ne puisse être créé, un certificat racine CA est nécessaire. Si aucun n'est trouvé, le script créera tout d'abord son propre certificat racine CA.

   ```
   cd /etc/zarafa/ssl/
   sh /usr/share/doc/zarafa/ssl-certificates.sh server
   ```

3. Saisir un mot (phrase) de passe si son utilisation est souhaitée avec la clé du serveur. Si un mot de passe est défini, c'est ce mot de passe qui sera ensuite utilisé pour signer les requêtes de certificat. Puis renseigner les données du certificat. Faire attention au champ 'Common Name'. Ce doit être le FQDN du serveur. Le champ 'Challenge Password' à la fin, peut être laissé vide. Au terme de sa création, le certificat devra être contresigné par le CA. Accepter deux fois de suite la demande de signature et saisir de nouveau le mot de passe du CA lorsque que celui-ci sera demandé.

4. En dernier lieu, le script demandera si la clé publique du certificat doit être affichée. Ceci n'est pas nécessaire puisque les certificats on déjà été créés.

5. Au terme de l'exécution du script **ssl-certificates.sh**, le certificat du serveur est créé dans le répertoire courant. Le certificat racine CA se trouve généralement dans le même répertoire ou dans le répertoire SSL par défaut de la distribution Linux. Sur Ubuntu le certificat racine CA sera créé en tant que **./demoCA/cacert.pem**, sur RedHat le certificat racine CA sera créé en tant que **/etc/CA/cacert.pem**. Modifier les lignes suivantes dans le fichier **/etc/zarafa/server.cfg**.

```
server_ssl_enabled            = yes
server_ssl_port               = 237
server_ssl_ca_file            = /etc/zarafa/ssl/demoCA/cacert.pem
server_ssl_key_file           = /etc/zarafa/ssl/server.pem
server_ssl_key_pass           = <ssl-password>
sslkeys_path                  = /etc/zarafa/sslkeys
```

6. Après un redémarrage du service, Zarafa-server devrait accepter les connexions HTTPS. Veuillez vérifier le fichier de journalisation du serveur pour déceler toute erreur potentielle.

7. Pour plus d'options concernat les certificats SSL, veuillez consulter les pages man de zarafa-server.cfg.

8. Si les certificats du serveur ont été créés avec succès, les certificats des clients pourront être créés selon les étapes suivantes :

```
cd /etc/zarafa/ssl
sh /usr/share/doc/zarafa/ssl-certificates.sh client
```

9. Renseigner toutes les données, comme pour le certificat de serveur. Sur certaines distributions Linux, la valeur du champ Common Name ne peut être identique à celle du certificat serveur. Au terme de sa création, le certificat devra être contresigné par le CA et une clé publique devra lui être créée.

10. Deux certificats clients sont créés : **client.pem** et **client-public.pem**. Le fichier **client.pem** est la clé privée qui sera utilisée par un client (comme zaraga-dagent ou zarafa-spooler). Le fichier **client-public.pem** est la clé publique qui sera utilisée par le serveur.

11. Déplacer la clé publique vers le répertoire **/etc/zarafa/sslkeys**.

```
mv /etc/zarafa/ssl/client-public.pem /etc/zarafa/sslkeys
```

12. Redémarrer le service **zarafa-server** sur tous les nœuds afin d'activer tous les nouveaux certificats :

```
/etc/init.d/zarafa-server restart
```

13. Pour tester le certificat SSL client, modifier les lignes suivantes dans le fichier **/etc/zarafa/dagent.cfg**.

```
server_socket = https://127.0.0.1:237/zarafa
sslkey_file = /etc/zarafa/ssl/client.pem
sslkey_pass = <ssl-client-password>
```

Une fois que les certificats ont été configurés, le courrier peut ensuite être distribué en utilisant la connexion SSL avec la clé privée de dagent, dans cet exemple, sur l'hôte local.

```
zarafa-dagent -v -c /etc/zarafa/dagent.cfg <nom_utilisateur_sur_ce_noeud>
Sujet :  courriel test
Test
<ctrl-d>
```

Lors d'une connexion SSL, l'utilitaire dagent vérifiera la clé privé à l'aide du certificat racine CA. Sur les systèmes de type Red Hat, des fichiers d'empreinte cryptographique sont générés à partir des certificats racine CA :

```
yum install openssl-perl
cp /etc/CA/cacert.pem /etc/pki/tls/certs/zarafa-ca.pem
c_rehash
```

L'utilitaire dagent peut ainsi authentifier une clé privé à l'aide du trousseau. Sur les systèmes de type Debian, cette étape peut être ignorée.

14. Si le test s'effectue avec succès, il est possible de modifier la valeur suivante dans dagent.cfg :

```
server_socket = file:///var/run/zarafa
```

15. Déployer tous les certificats vers les différents nœuds multi-serveur :

```
scp -r /etc/zarafa/ssl /etc/zarafa/sslkeys root@node2:/etc/zarafa/
```

Ne pas oublier de copier le certificat racine CA vers chacun des différents nœuds si ce fichier est placé en dehors des répertoires qui viennent juste d'avoir été copiés.

16. Renouveler les étapes ci-dessus afin de configurer **`server.cfg`** et **`dagent.cfg`** sur chacun des différents nœuds. Sur les nœuds de type Red Hat il faut également ajouter le certificat racine CA au trousseau. Une fois terminé, vérifier la distribution à distance :

```
zarafa-dagent -v -c /etc/zarafa/dagent.cfg <nom_utilisateur_sur_autre_nœud>
Subject: courriel test
Test
<ctrl-d>
```

Cette action ne devrait pas occasionner de message d'erreur de livraison, sinon, veuillez vérifier les certificats qui viennent d'être créés. Il est désormais possible de distribuer du courrier à partit d'un MTA centralisé vers les différents nœuds multi-serveur.

Les certificats SSL clients peuvent être utilisés par les utilitaires suivants pour se connecter à un serveur Zarafa distant :

```
zarafa-dagent
zarafa-spooler
zarafa-backup, zarafa-restore
zarafa-admin
```

Dans le cas d'environnements multi-serveur élaborés et pour bénéficier de la meilleure configuration qui soit adaptée à un contexte spécifique, les services professionnels de Zarafa sont disponibles pour tout conseil et support technique.

6.4. Utilitaire de mise à jour du Client Windows de Zarafa

ZCP comporte un mécanisme permettant aux clients Windows de Zarafa de se mettre à jour vers la version la plus récente.

Note

L'utilitaire de mise à jour du client Windows de Zarafa n'est disponible que pour les éditions ZCP Professional ou ZCP Entreprise.

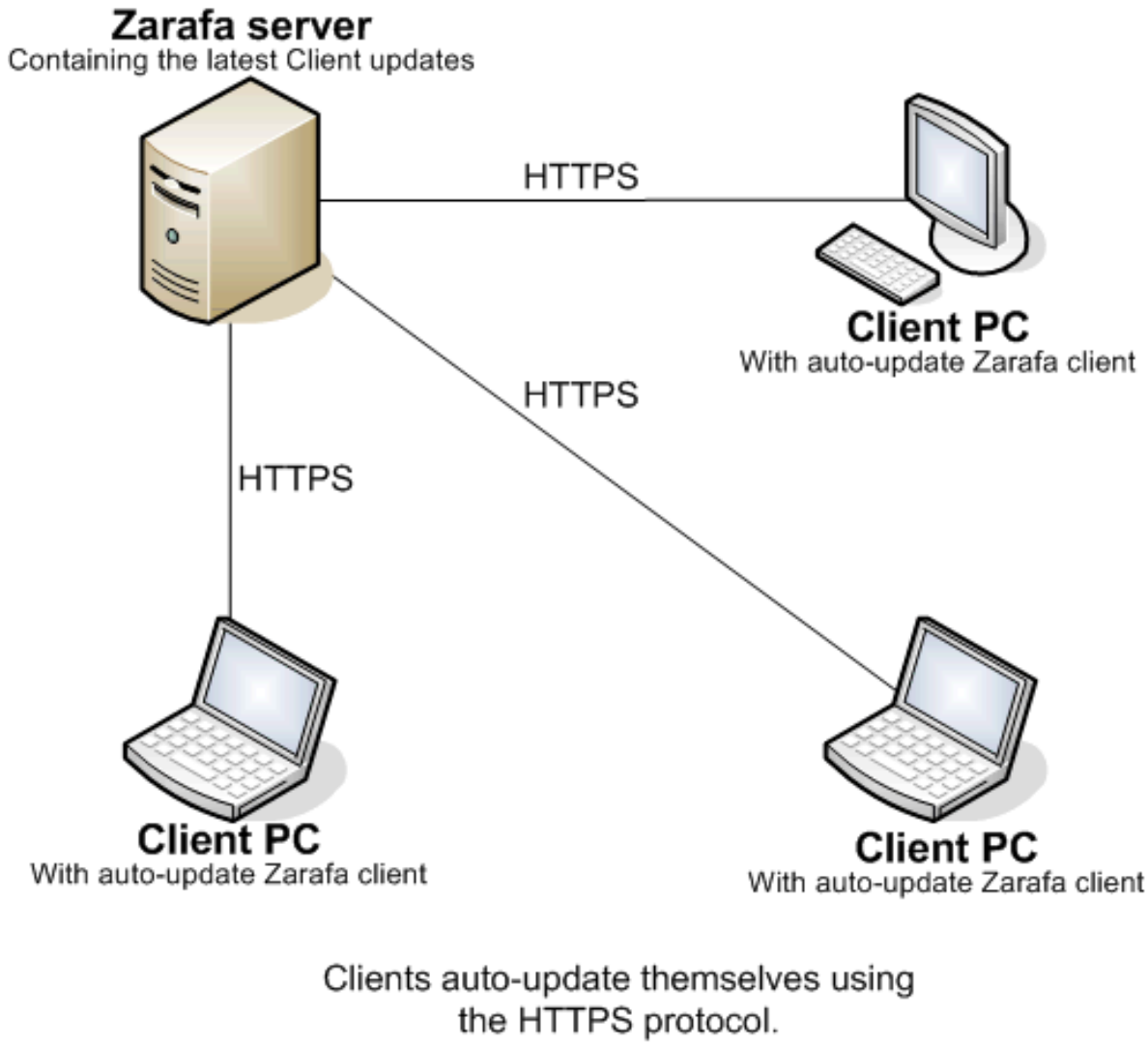


Figure 6.8. Structure de la mise à jour automatique

Restrictions :

- Le mécanisme de mise à jour automatique n'est pas en mesure de rétrograder le client vers une version inférieure, il mettra toujours le client Zarafa à jour vers la version la plus récente disponible.
- L'utilitaire de mise à jour du client Windows de Zarafa n'est pas disponible pour Windows 2000 ou une version inférieure

6.4.1. Configuration du serveur

L'activation de la mise à jour automatique du client Windows de Zarafa s'effectue en définissant les paramètres suivants sur **yes** dans le fichier **server.cfg** du serveur Zarafa:

```
client_update_enabled = yes
```

Lors de la mise à jour de **zarafa-server**, la dernière version du programme d'installation du client sera copiée à l'emplacement spécifié dans le fichier de configuration **server.cfg**, comme indiqué ci-dessous.

```
client_update_path = /var/lib/zarafa/client
```

Le client de mise à jour automatique peut renvoyer des données de journalisation au serveur
Si l'utilitaire de mise à jour rencontre un échec, alors les fichiers de journalisation sera envoyé au serveur principal. Ce comportement peut être modifié à l'aide du paramétrage suivant :
client_update_log_level = 1

Les options suivantes peuvent être définies : 0 - désactivé, 1 - envoyer les fichiers de journalisation au serveur uniquement en cas d'erreur, 2 - toujours envoyer les fichiers de journalisation au serveur

Les fichiers de journalisation envoyés par le client de mise à jour automatique sont placés sur le serveur dans le répertoire suiovant : client_update_log_path = /var/log/zarafa/autoupdate

Dans le dossier de mise à jour du client, les mises à jour observent une stricte nomenclature. Le serveur Zarafa ne fonctionnera qu'avec les mises à jour adhérant à cette nomenclature :

```
zarafaclient-<major version>.<minor version>.<update number>-<build number>.msi
```

Par exemple **zarafaclient-6.40.0-19050.msi** est un nom valide de mise à jour.

C'est sur la base de cette nomenclature que l'utilitaire de mise à jour du client Windows de Zarafa détermine la disponibilité d'une nouvelle version du logiciel client. Si un client émet une requête de mise à jour vers une nouvelle version, **zarafa-server** lui renverra le dernier package de mise à jour et le client pourra ainsi effectuer lui-même sa propre mise à jour vers la version la plus récente disponible.

Note

Si le profil par défaut est configuré pour l'utilisation du chiffrement par le port **237**, le certificat CA doit être installé sur la machine utilisée.

6.4.2. Configuration du client

Le mécanisme de mise à jour automatique du client Windows de Zarafa est composé d'une application qui démarre le processus de mise à jour et qui s'intitule **ZarafaLaunchUpdater.exe** ainsi que d'un service Windows nommé **ZarafaUpdaterService.exe**.

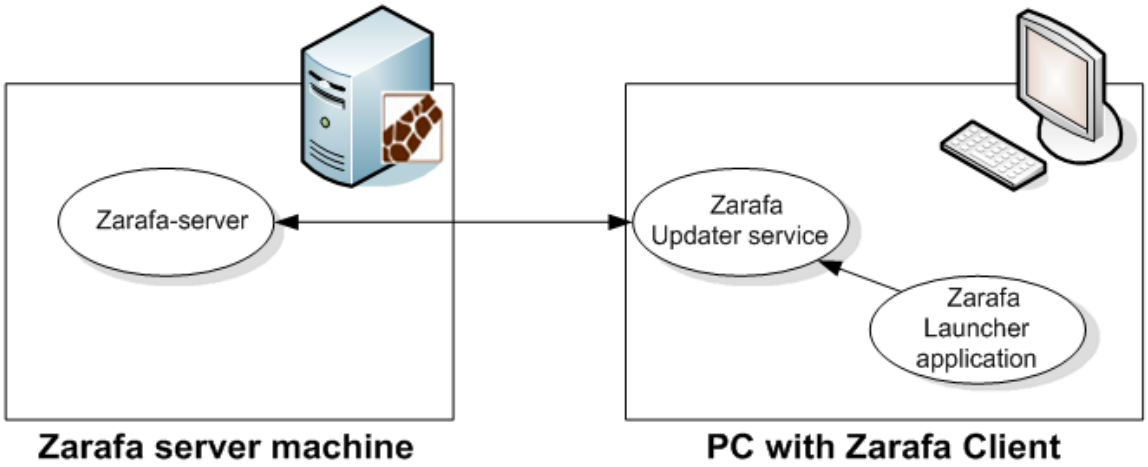


Figure 6.9. Structure de la mise à jour automatique

L'utilitaire de lancement de la mise à jour sera exécuté au démarrage de Windows. La commande permettant de lancer l'application est placée dans le registre :

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
```

Cette application déterminera la version actuelle du client à partir de la clé de registre suivante :

```
HKEY_LOCAL_MACHINE\Software\Zarafa\Client\Version
```

Cette clé de registre contient la version actuelle du client Windows de Zarafa installé sur la machine.

L'utilitaire de lancement de la mise à jour consultera le profil par défaut d'Outlook dans la base de registre afin de réunir les informations nécessaires permettant de se connecter au serveur Zarafa. Il transmettra la version actuelle du client Windows de Zarafa au serveur Zarafa qui répondra par l'envoi d'un version plus récente du client Windows de Zarafa si disponible.

6.4.2.1. Service de mise à jour Zarafa

Le service de mise à jour Zarafa est lancé en tant qu'utilisateur local. Il possède donc tous les privilèges nécessaires pour lui permettre d'installer le client Windows de Zarafa sur le bureau.

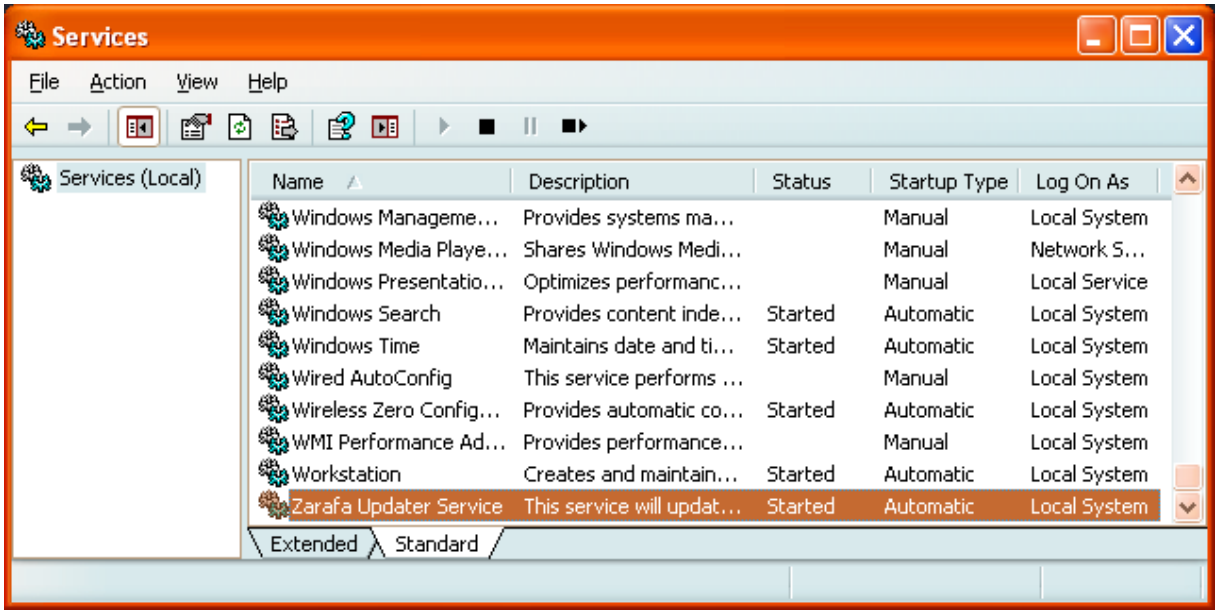


Figure 6.10. Services

Le service de mise à jour de Zarafa attendra le transfert d'information provenant de l'application de lancement de la mise à jour lui fournissant la version actuelle du client ainsi que les coordonnées du serveur Zarafa auquel se connecter Si une mise à jour adéquate est disponible, le service la téléchargera vers l'emplacement **`c:\windows\temp\zarafaclient.msi`**. Le service de mise à jour Zarafa lancera l'installation en mode silencieux

Bien que le processus entier soit effectué en mode silencieux, des fichiers de journalisation seront générés pour un éventuel diagnostic de dépannage. La journalisation du service de mise à jour sera tenue dans le répertoire **`All users\Application data\`** et la journalisation de l'application de lancement de la mise à jour sera tenue dans le répertoire **`<user>\Application data\`** directory.

Lorsque le service de mise à jour démarre la mise à jour d'un client, il crée alors les fichiers zarafa-<trackid>.log et zarafa-<trackid>msi.log dans le répertoire **`<user>\Local Settings\Temp`**. Ces fichiers sont ensuite envoyés au serveur qui à été défini dans la configuration.

Note

Le client ne pourra trouver les mises à jour avec succès que si le profil par défaut d'Outlook est configuré pour fonctionner avec le serveur Zarafa, et si des mises à jour sont disponibles sur ce même serveur. Même si les paramètres sont définis avec la valeur *`prompt for the profile to be used'* (demander le profil qui doit être utilisé) le gestionnaire de mise à jour du client Windows de Zarafa s'exécutera avec succès si le menu déroulant (désactivé) spécifie le profil qui a été défini pour Zarafa. Veuillez vous référer au manuel de l'utilisateur pour la configuration des profils Outlook.

6.4.2.2. Statut de Zarafa Updater

`zarafa-server` retourne le statut de l'utilitaire de mise à jour du client Zarafa dans le fichier **`server.log`**. **`zarafa-admin`** retourne le dernier statut de la mise à jour du client. Vous pouvez afficher les informations de mise à jour par utilisateur à l'aide de la commande suivante : zarafa-admin --details <user>

+

```
 Client update Information:
Trackid:                1889610488
Last update:            <date>
From version:           <version>
To version:             <version>
Computername:           <name>
Update:                 Succeed
```

Lorsque la mise à jour d'un client échoue, les fichiers de journalisation sont disponibles dans le répertoire qui a été défini dans le champ **`client_update_log_path`** du fichier de configuration **`server.cfg`** (paramétré par défaut sur /var/log/zarafa/autoupdate). La valeur de trackid peut être utilisée afin de retrouver les fichiers de journalisation, par exemple : /var/log/zarafa/autoupdate/0x70A12AF8/

+

zarafa-autoupdate.log zarafa-msi.log

6.4.3. Options MSI

Si vous souhaitez plutôt initier l'installation de votre client Zarafa à partir de votre serveur de domaine Windows, il est préférable de pas installer Zarafa Updater Service. Dans ce cas, il faut configurer les options suivantes :

ADDDEFAULT="Client"
: Ainsi, seul le client Outlook sera installé, et non pas le service de mise à jour. Pour installer également ce service, il faut ajouter "Updater" à l'option.

APPDIR=D:\Zarafa\Client
: Pour modifier le chemin d'installation par défaut, utiliser la variable APPDIR. Si cette option n'est pas modifiée, le répertoire "Program Files" sera utilisé.

/q
: Rendre l'installation silencieuse. Aucune interface graphique ne s'affichera. Pour afficher les progrès de l'installation, utiliser l'option b (pour interface graphique de base) ou r (pour interface graphique réduite). Si vous affichez l'interface complète (option f), elle sera interactive.

Exécuter **msiexec** pour consulter la liste de toutes les options disponibles. Pour lancer une installation automatisée classique, exécuter la commande suivante :

```
msiexec /i zarafaclient-en.msi ADDDEFAULT=Client /q
```

Note

Pour une installation automatisée, vous devez utiliser le fichier zarafaclient-en.msi. Cet installeur est disponible uniquement en anglais et a été créé spécifiquement pour cette tâche.

6.5. Utiliser les services ZCP avec les privilèges d'un utilisateur standard

Normalement les services Zarafa sont lancés par root. Depuis la version 5.0, il est possible de changer l'utilisateur exécutant le service, tout en continuant à démarrer les services en tant que root. Cependant, plusieurs étapes doivent être réalisées avant que des services puissent correctement être exécutés en tant qu'utilisateur non-root.

Si **log_method** est défini sur **file**, il faut s'assurer que le dossier et le fichier soient accessibles en écriture à l'utilisateur ou au groupe qui exécutera le service. Si la commande logrotate est exécutée, en envoyant le signal HUP au service, un nouveau fichier sera créé et appartiendra à l'utilisateur exécutant le service.

Le service devra toujours être démarré en tant que root puisqu'il crée un fichier pid à l'emplacement **/var/run**, et qu'il ouvrira des connexions réseau utilisant probablement des ports inférieurs à **1024**, qui ne peuvent être ouverts que par **root**.

L'exemple suivant montre comme configurer **zarafa-server** afin qu'il fonctionne sous l'utilisateur **zarafa** et le groupe **zarafa**:

```
addgroup --system zarafa
adduser --system ---home /dev/null ---no-create-home \
       --ingroup zarafa \
       --disabled-password --gecos 'Zarafa services' \
       --shell /bin/false zarafa
mkdir /var/log/zarafa
chown zarafa:zarafa /var/log/zarafa
chown zarafa:zarafa /etc/zarafa/report
chown -R zarafa:zarafa /var/lib/zarafa
```

Note

Les utilitaires addgroup et adduser peuvent utiliser différentes syntaxes selon les distributions.

Modifier les options **run_as_user** et **run_as_group** dans le fichier **server.cfg** et les définir toutes deux sur zarafa. S'assurer que l'option **local_admin_users** conserve toujours root en tant qu'administrateur, afin de pouvoir continuer à utiliser l'utilitaire **zarafa-admin**. Autrement **su** ou **sudo** devront être utilisés chaque fois que l'utilitaire **zarafa-admin** sera lancé.

6.6. Authentification unique (SSO) avec ZCP

Ce chapitre présente la mise en place d'un environnement d'authentification unique avec ZCP, afin que les utilisateurs puissent être authentifiés sans avoir à fournir de mot de passe. ZCP prend en charge les protocoles d'authentification NTLM et Kerberos. La prise en charge de Kerberos est disponible à partir de la version 6.40.2 ou supérieure de ZCP.

Les deux méthodes sont détaillées dans les sections suivantes.

6.6.1. Authentification unique NTLM avec ADS

6.6.1.1. Installation des logiciels Linux

Les logiciels suivants doivent être installés :

- **winbind**
- **kinit**

Selon votre distribution Linux, ces logiciels peuvent être fournis par des packages portant des noms différents. Sur une distribution de type Debian:

```
apt-get install krb5-user winbind
```

krb5-user installera également les fichiers de configuration de la bibliothèque Kerberos dans le répertoire **/etc**. Le package **winbind** dépend de **samba-common** qui sera donc également installé. Sur Red Hat Enterprise Linux, les packages **krb5-workstation** et **samba-common** sont tous les deux requis.

Pour activer l'authentification unique NTLM avec ZCP, configurer l'option suivante dans le fichier **server.cfg** :

```
enable_sso  = yes
```

6.6.1.2. ADS : paramètres spécifiques de réseau

Les conditions préalables suivantes doivent être remplies avant de pouvoir continuer :

- Chaque serveur doit avoir un nom de DNS, afin que leur adresse IP puisse-t-être trouvée par DNS.
- Toutes les horloges des serveurs doivent être synchronisées. Il ne doit y avoir aucune latence, ne serait-ce que de quelques minutes.

Ce document utilise les noms suivants comme exemples:

- **FQDN** du serveur ADS Windows : **ADSSERVER.ADSDOMAIN.LOCAL**. Par conséquent, votre serveur Windows sera nommé : **ADSSERVER**, le royaume sera **ADSDOMAIN.LOCAL**, et le nom du domaine sera **ADSDOMAIN**. Les stations de travail pourront donc joindre votre domaine en utilisant soit le nom **ADSDOMAIN** ou soit le nom **ADSDOMAIN.LOCAL** name.
- **FQDN** du serveur Linux sera **LINUXSERVER.LOCAL**. Ce nom importe peu à condition qu'il soit géré par le serveur DNS.

6.6.1.3. Configuration de la bibliothèque Kerberos

Il faut premièrement configurer la bibliothèque Kerberos. Le fichier de configuration est **/etc/krb5.conf**. Sous la section **libdefaults**, définir :

```
default_realm = ADSDOMAIN.LOCAL
```

Sous la section **realms**, ajouter le domaine Windows :

```
[realms]
ADSDOMAIN.LOCAL = {
        kdc = 192.168.0.100
        admin_server = 192.168.0.100
        password_server = 192.168.0.100
        default_domain = ADSDOMAIN.LOCAL
}
```

Ici, **192.168.0.100** est l'adresse IP de votre serveur de domaine ADS Windows.

La bibliothèque Kerberos est maintenant configurée, il est désormais possible de s'identifier à l'aide de **kinit** sur le serveur Linux :

```
Administrateur kinit
```

Ceci nécessite un mot de passe :

```
Mot de passe de Administrator@ADSDOMAIN.LOCAL :
```

Renseigner le mot de passe administrateur, et un ticket Kerberos devrait être fourni par le serveur ADS.

6.6.1.4. Se connecter à un domaine ADS

Il faudra premièrement configurer Samba. Ouvrir le fichier **`/etc/samba/smb.conf`**, et ajouter ou modifier les options de configurations suivantes :

For Samba < 3.4

```
[global]
realm = ADSDOMAIN.LOCAL
use kerberos keytab = true
security = ads
```

For Samba >= 3.4

```
[global]
realm = ADSDOMAIN.LOCAL
kerberos method = dedicated keytab
dedicated keytab file = /etc/krb5.keytab
security = ads
```

The value of **`kerberos method`** may also be set to **`system keytab`**, and **`dedicated keytab file`** may be left out. Please consult the **`smb.conf(5)`** manual page for more information about these settings.

Grâce au ticket précédemment reçu, il sera possible de se connecter au domaine Windows sans avoir de nouveau à fournir le mot de passe :

```
net ads join
```

ou si ça ne fonctionne pas :

```
net ads join -S ADSDOMAIN -U Administrator
```

Cette commande peut différer selon les versions de Samba. Si cette commande vous demande un mot de passe, c'est qu'une erreur s'est produite et il faudra terminer le programme à l'aide des touches CTRL + C. Si tout se déroule correctement, la ligne suivante doit s'afficher à l'écran :

```
Joined 'LINUXSERVER' to realm 'ADSDOMAIN.LOCAL'
```

ou un quelconque autre message de succès similaire.

Il faudra maintenant redémarrer le daemon winbind, parce qu'il conserve trop d'éléments en cache :

```
/etc/init.d/winbind restart
```

C'est tout. Pour vérifier si l'authentification fonctionne correctement, veuillez exécuter les commandes suivantes :

```
ntlm_auth --username=john
```

En supposant que **`john`** soit un utilisateur sur votre serveur ADS.

Le programme demandera alors un mot de passe. Une fois le mot de passe renseigné, le message suivant devrait s'afficher :

```
NT_STATUS_OK: Success (0x0)
```

Si cette étape échoue, essayer de redémarrer **winbind**, vérifier les noms DNS, vérifier à l'aide de **strace** ce que **ntlm_auth** tente de faire, vérifier à l'aide de **tcpdump** si une connexion existe réellement sur votre réseau entre **ntlm_auth** et le serveur de domaine, et essayer d'autres utilitaires de résolution d'erreur de bas niveau.

6.6.2. Authentification unique NTLM avec Samba

6.6.2.1. Installation des logiciels Linux

Les logiciels suivants doivent être installés sur le serveur ZCP:

```
winbind
```

Selon votre distribution Linux, ce logiciel peut être fournis par au package portant un nom différent. Sur une distribution de type Debian:

```
apt-get install winbind
```

Sur Red Hat Enterprise Linux, le package **samba-common** est requis.

Pour activer l'authentification unique NTLM avec ZCP, définir l'option suivante dans le fichier de configuration **server.cfg** :

```
enable_sso  = yes
```

6.6.2.2. Connexion au domaine

Ensuite, il sera possible de se connecter au domaine Samba à l'aide de la commande suivante :

```
net rpc join
```

Finalement, il faudra saisir le mot de passe administrateur. Si l'opération se déroule avec succès, l'invite de commande affichera le message suivant :

```
Joined domain <DOMAIN>
```

La configuration de l'authentification unique est maintenant achevée. Pour vérifier si l'authentification fonctionne correctement, veuillez exécuter les commandes suivantes :

```
ntlm_auth --username=john
```

En supposant que **john** soit un utilisateur Samba valide.

Le programme demandera alors un mot de passe. Une fois le mot de passe renseigné, le message suivant devrait s'afficher :

```
NT_STATUS_OK: Success (0x0)
```

Si cette étape échoue, essayer de redémarrer **winbind**, vérifier les noms DNS, vérifier à l'aide de **strace** ce que **ntlm_auth** tente de faire, vérifier à l'aide de **tcpdump** si une connexion existe réellement sur votre réseau entre **ntlm_auth** et le serveur de domaine, et essayer d'autres utilitaires de résolution d'erreur de bas niveau.

6.6.3. Authentification unique avec Kerberos

6.6.3.1. Conditions requises et conventions

- MIT Kerberos doit être installé sur le serveur qui exécute ZCP.
- La version 6.40.2 ou supérieure de ZCP doit être installée pour pouvoir bénéficier de l'authentification unique avec Outlook.
- Chaque serveur doit avoir un nom de DNS, afin que leur adresse IP puisse-t-être trouvée par DNS. Il est également nécessaire que tous les serveurs possèdent un enregistrement PTR.
- Tous les horloges des serveurs doivent être synchronisées.

Ce document utilise les noms suivants comme exemples:

- FQDN du serveur Active Directory de Windows : **ADSERVER.ADSDOMAIN.LOCAL**. Par conséquent, votre serveur Windows sera nommé : **ADSERVER**, le domaine sera **ADSDOMAIN.LOCAL**, et le nom du groupe de travail sera **ADSDOMAIN**.
- FQDN du serveur Zarafa sera **ZARAFA.LINUXDOMAIN.LOCAL**.

Dans cet exemple le serveur Zarafa est placé dans un domaine différent. Ce n'est pas une condition requise, mais cela clarifie les instructions détaillant la création du principal Kerberos.

6.6.3.2. Configuration Active Directory

Créer deux principaux Kerberos dans Active Directory, un pour l'authentification unique avec WebAccess et un pour l'authentification unique avec Outlook.

1. Ajouter un nouvel utilisateur **httpd-linux** à Active Directory (cet utilisateur servira à créer le principal pour l'authentification unique avec WebAccess, l'identifiant de cet utilisateur peut être différent).
2. Ajouter un nouvel utilisateur **zarafa-linux** à Active Directory (cet utilisateur servira à créer le principal pour l'authentification unique avec Outlook, l'identifiant de cet utilisateur peut être différent).
3. S'assurer que l'option *Password never expires* est activée.
4. Activer : *Use DES encryption types for this account* dans les propriétés des comptes de ces deux utilisateurs.
5. Après avoir effectué la configuration de cette propriété, il est hautement recommandé de redéfinir les mots de passes de ces deux utilisateurs.

Installer les utilitaires de support Windows comportant le programme **ktpass.exe** sur le serveur Active Directory. Ces utilitaires de support se trouvent sur le CD d'installation de Windows Server ou peuvent être téléchargés à partir du site Web de Microsoft.

Note

Au cours de la création d'un fichier keytab sur Windows Server 2008, s'assurer de spécifier **RC4-HMAC-NT** comme type de chiffrement, *-mapop set +desonly* ne doit pas être configuré.

Exécuter les commandes suivantes afin de créer le fichier keytab pour le serveur Web Apache :

```
ktpass.exe -princ HTTP/zarafa.linuxdomain.local@ADSDOMAIN.LOCAL
 -mapuser EXAMPLE\httpd-linux -crypto DES-CBC-MD5 -ptype KRB5_NT_PRINCIPAL
 -mapop set +desonly -pass <password> -out c:\keytab.apache
```

Exécuter les commandes suivantes afin de créer le fichier keytab pour le serveur Zarafa :

```
ktpass.exe -princ zarafa/zarafa.linuxdomain.local@ADSDOMAIN.LOCAL
 -mapuser EXAMPLE\zarafa-linux -crypto DES-CBC-MD5 -ptype KRB5_NT_PRINCIPAL
 -mapop set +desonly -pass <password> -out c:\keytab.zarafa
```

- Copier le fichier **keytab.apache** vers le répertoire **/etc/httpd/conf/** du serveur Linux.
- Copier le fichier **keytab.zarafa** vers le répertoire **/etc/zarafa/** du serveur Linux.

6.6.3.3. Configuration Kerberos

Ouvrir le fichier **/etc/krb5.conf** et y insérer les lignes suivantes :

```
[libdefaults]
       default_realm = ADSDOMAIN.LOCAL
       default_tgs_enctypes = des-cbc-md5 arcfour-hmac-md5
       default_tkt_enctypes = des-cbc-md5 arcfour-hmac-md5
       permitted_enctypes = des-cbc-md5 arcfour-hmac-md5

[realms]
       ADSDOMAIN.LOCAL = {
               kdc = adserver.adsdomain.local
               admin_server = adserver.adsdomain.local
       }

[domain_realm]
 .adsdomain.local = ADSDOMAIN.LOCAL
 adsdomain.local = ADSDOMAIN.LOCAL
```

Configurer ZCP pour une authentification unique Kerberos avec Outlook

Ajouter la ligne suivante à la section **[libdefaults]** du fichier **/etc/krb5.conf**:

```
default_keytab_name = /etc/zarafa/keytab.zarafa
```

6.6.3.4. Configuration du serveur Zarafa

Pour activer l'authentification unique Outlook avec ZCP, définir l'option suivante dans le fichier de configuration **server.cfg** :

```
enable_sso = yes
```

Si le nom d'hôte du serveur Linux (voir la commande **hostname**) n'est pas identique au FQDN du serveur Linux, la variable **server_hostname** devra être modifiée dans le fichier **server.cfg** :

```
server_hostname = zarafa.linuxdomain.local
```

Redémarrer zarafa-server afin d'activer toutes les modifications.

```
service zarafa-server restart
```

6.6.3.5. Configuration Apache (pour l'authentification unique avec WebAccess)

Installer le module Apache **mod_auth_kerb**, p. ex. pour Red Hat :

```
yum install mod_auth_kerb
```

Ouvrir le fichier /etc/httpd/conf.d/auth_kerb.conf. Ajouter les lignes suivantes à la fin de ce fichier :

```
Alias /webaccess /usr/share/zarafa-webaccess

<Directory /usr/share/zarafa-webaccess>
  AuthType Kerberos
  AuthName "Kerberos Login"
  KrbMethodNegotiate On
  KrbMethodK5Passwd Off
  KrbServiceName HTTP
  KrbAuthRealms ADSDOMAIN.LOCAL
  Krb5KeyTab /etc/httpd/conf/keytab.apache
  require valid-user
</Directory>
```

Définir les permissions du fichier keytab sur 400 et attribuer sa propriété à l'utilisateur Apache :

```
chmod 400 /etc/httpd/conf/keytab.apache
chown apache:apache /etc/httpd/conf/keytab.apache
```

Redémarrer le service Apache afin d'activer toutes les modifications, p. ex. pour Red Hat :

```
service httpd restart
```

6.6.3.6. Configuration WebAccess

Pour mettre en place un environnement d'authentification unique avec la plateforme collaborative Zarafa Collaboration, il est nécessaire d'établir un niveau de confiance entre le serveur Web Apache et le serveur de stockage Zarafa. Ce niveau de confiance est nécessaire afin de gérer l'authentification de WebAccess par le serveur Web Apache et non plus par le serveur de stockage Zarafa.

Pour créer ce niveau de confiance, ajouter l'utilisateur exécutant Apache à la ligne suivante dans le fichier **/etc/zarafa/server.cfg** :

```
local_admin_users = root apache
```

Pour configurer l'authentification unique dans Zarafa WebAccess, modifier l'option suivante dans le fichier **config.php** :

```
define("LOGINNAME_STRIP_DOMAIN", true);
```

Note

Dans cette configuration nous présumons que Zarafa WebAccess est installé sur le même serveur que le serveur de stockage Zarafa.

Redémarrer zarafa-server afin d'activer la modification, p. ex. pour Red Hat :

```
service zarafa-server restart
```

6.6.3.7. Configuration du navigateur Internet

Avant que l'authentification unique ne puisse être utilisée dans un navigateur Internet, il faut configurer les options suivantes :

Firefox

1. Saisir **`about:config`** dans la barre d'adresse
2. Filtrer sur **`auth`**
3. Modifier les options : **`network.negotiate-auth.trusted-uris`** et **`network.negotiate-auth.delegation-uris`** avec la valeur **`.testdomain.com`**

Internet Explorer

1. Aller sur *Outils* **>** *Options Internet* **>** *Avancé*
2. S'assurer que l'option *Activer l'authentification intégrée de Windows* soit cochée
3. Ajouter l'URL du serveur Zarafa (*http://zarafa.linuxdomain.local*) aux sites *Local Intranet*.

Redémarrer le navigateur et ouvrir WebAccess via le FQDN (*http://zarafa.linuxdomain.local/webaccess*). Si la configuration a correctement été effectuée, l'utilisateur sera connecté à WebAccess sans avoir à saisir son identifiant ni son mot de passe.

6.6.4. Fonctionnement

Une fois que l'authentification unique (SSO) fonctionne correctement sur votre serveur Linux, elle sera utilisée automatiquement par **`zarafa-server`**. S'identifier sur une station de travail Windows et créer un nouveau profil Outlook pour l'utilisateur avec lequel vous venez de vous identifier, mais laisser le champ du mot de passe vide. Outlook créera alors le profil sans mot de passe.

6.7. Suivi des messages avec Zarafa Archiver

Cette section comporte des informations sur les suivi de tous les messages entrants et sortants à l'aide de la nouvelle technologie d'archivage de Zarafa. Cette procédure est utile dans les environnements plus stricts de messagerie électronique où il est important de vérifier tout ce qui a été envoyé et reçu malgré la manière dont le *propriétaire* des messages en a disposé.

6.7.1. Archivage à la distribution

L'archivage à la distribution consiste à s'assurer que chaque message reçu soit également placé dans le dossier d'archivage adéquat. Si le message ne peut pas être archivé, il **ne sera pas** distribué. Ceci

occasionnera une défaillance temporaire, et le MTA essayera à nouveau de distribuer le message plus tard.

L'archivage à la distribution est géré par le service **`zarafa-dagent`** et peut être contrôlé à l'aide de l'option **`archive_on_delivery`** dans le fichier de configuration de ce service.

Lorsqu'un message est archivé avec cette méthode, un archivage ordinaire est généré, ce qui signifie que les règles normales lui sont appliquées. Cela signifie que si l'utilisateur déplace le message dans la base de stockage principale, le message sera également déplacé dans l'archivage. Ceci inclut également tout déplacement vers la corbeille.

Important

Lorsqu'un message est supprimé de la base de stockage principale, le message **n'est pas** supprimé de l'archive. Au lieu de cela, il est déplacé vers le dossier spécial 'Deleted' dans l'archiver.

6.7.2. Archivage à l'envoi

L'archivage à l'envoi consiste à s'assurer que chaque message envoyé soit également placé dans le dossier d'archivage adéquat. Si le message ne peut pas être archivé, il **ne sera pas** envoyé. Au lieu de cela, un message d'échec sera envoyé à l'utilisateur.

L'archivage à la distribution est géré par le service **`zarafa-spooler`** et peut être contrôlé à l'aide de l'option **`archive_on_send`** dans le fichier de configuration de ce service.

Important

Le courriel envoyé directement à un serveur SMTP (généralement avec l'utilisation d'un compte IMAP) ne sera pas archivé directement puisque dans ce cas, **`zarafa-spooler`** ne prend pas part au processus d'envoi.

Lorsqu'un message est archivé avec cette méthode, un archivage détaché est généré. Cela signifie que l'archive ne garde aucun lien avec le message d'origine dans la base de stockage principal. De même, aucun message dans la base de stockage principale ne garde de lien avec le message archivé.

Note

À moins que l'option ne soit désactivée, les messages du dossier 'Sent Items' sont archivés comme n'importe quel autre message. Un espace de stockage supplémentaire est nécessaire puisque ces messages ont également été mis en archive par zarafa-spooler.

6.8. Zarafa Python plugin framework

The Zarafa Spooler and the Zarafa Dagent support the Zarafa python plugin framework. This framework makes it easier for advanced system administrators and developers to integrate systems with the spooler and dagent. The advanced system administrator and developer can easily add new functionality or change some behaviours of the existing system. The plugin framework is based on the programming language Python which means that you need to create your own hook in python.

6.8.1. How it works

If the plugin framework in the spooler or dagent is enabled it will search for python files in the directory **plugin_path** and look for a specific type of plugin. If the plugins are found it will be verified and loaded. Everytime the spooler or dagent is called it will execute the hooks based on priority. Plugins can validate and change a message on different stages of the spooler and dagent process.

6.8.2. General Options

The options for the python plugin framework are for every client the same except the file locations, see *Tableau 6.1, « Table Python plugin framework options »*

Tableau 6.1. Table Python plugin framework options

Option	Default	Description
plugin_enabled	yes	Enable the plugin framework in the specfic component
plugin_manager_path	/usr/share/zarafa-**<componentname>**/python	Path to the plugin manager.
plugin_path	/var/lib/zarafa/**<componentname>**/plugins	Path to the activated plugins.

The value **<componentname>** can be *dagent* or *spooler*

6.8.3. How to use

After the installation of the component zarafa-dagent or zarafa-spooler it's possible to activate a plugin. The default plugins are installed in the folder '/usr/share/zarafa-**<componentname>**/python/plugins/'. To activate a plugin create a symbolic link in the **plugin_path** directory to the plugin which you want to activate. For example, to activate the disclaimer plugin in the spooler, run the follow command:

```
ln -s /usr/share/zarafa-spooler/python/plugins/disclaimer.py /var/lib/zarafa/spooler/plugins/
disclaimer.py
```

6.8.4. Zarafa-DAgent plugins

6.8.4.1. Move to public

The move to public plugin moves incoming messages to a folder in the public store.

Enable the move to public plugin, run the following command:

```
ln -s /usr/share/zarafa-dagent/python/plugins/movetopublic.py /var/lib/zarafa/dagent/plugins/
movetopublic.py
```

For this plugin is a config file required. Make a copy of the configuration file with the following command:

```
cp /usr/share/zarafa-dagent/python/plugins/movetopublic.cfg /etc/zarafa/movetopublic.cfg
```

6.8.4.2. BMP2PNG converter

The BMP2PNG plugin converts a BMP to PNG in the incoming email. This plugin can be used to reduce the image size of the delivered email.

Enable the BMP2PNG plugin, run the following command:

```
ln -s /usr/share/zarafa-dagent/python/plugins/BMP2PNG.py /var/lib/zarafa/dagent/plugins/
BMP2PNG.py
```

Note

The package **`python-imaging`** is required to use this plugin.

6.8.5. Zarafa-Spooler plugins

6.8.5.1. Disclaimer

The disclaimer plugin add a disclaimer to every email sent with the Zarafa spooler.

The disclaimer plugin supports plain text and HTML emails. RTF emails are not supported. To use the disclaimer plugin it's necessary to create the directory */etc/zarafa/disclaimers* which must include the disclaimers. The plugin is using the following files for the disclaimer:

Tableau 6.2. Table Disclaimer files

Filename	Description
default.txt	The plain text version of the disclaimer
default.html	The HTML version of the disclaimer
<companyname>.txt	The plain text version of the disclaimer of a company
<companyname>.html	The HTML version of the disclaimer of a company

Important

All files must encoded in utf8

Enable the disclaimer plugin, run the following command:

```
ln -s /usr/share/zarafa-spooler/python/plugins/disclaimer.py /var/lib/zarafa/spooler/plugins/
disclaimer.py
```

6.8.6. Troubleshooting

How to troubleshoot issues you might have while installing or using the Python plugin framework in the Zarafa dagent and spooler.

6.8.6.1. Log explanation

The Python plugin framework can log a lot of information so if there are issues it's recommended to set the **log_level** to 6. This will show all the information about the plugin framework.

Python error: No module named mapiplugin

The path to the plugin manager is invalid, this means the plugin framework can not be loaded and will result in the following error:

```
<DATE> [id] PYTHONPATH = /usr/share/zarafa-dagent/python/Unknown_path
<DATE> [id]   Python type: (null)
<DATE> [id]   Python error: No module named mapiplugin
<DATE> [id] Unable to initialize the dagent plugin manager
```

Check the path in **plugin_manager_path** should refer to the directory with the following files,

- mapiplugin.py
- pluginmanager.py
- plugintemplates.py
- wraplogger.py

Plugins not loaded

The path to the plugins directory is invalid or the permissions on the directory are invalid if this is the case you will receive the following error:

```
<DATE> [id] * Loading plugins started
<DATE> [id] ! Plugins directory '/usr/share/zarafa-dagent/python/plugins/invalid' doesn't
exists. Plugins not loaded.
```

Check the path in **plugin_path** by default it refer to the directory '/var/lib/zarafa/dagent/plugins', the permissions on the directory must atleast have read and execute permissions.

Python error: *PySwigObject* object has no attribute *Log*

There is an invalid version of MAPICore loaded. The old beta python-MAPI package installed the files in another directory but after removing the package the generated files are not removed after you start the dagent or spooler the old generated file is loaded an cause the following error:

```
<DATE> [id] PYTHONPATH = /usr/share/zarafa-dagent/python/
<DATE> [id]   Python type: (null)
<DATE> [id]   Python error: 'PySwigObject' object has no attribute 'Log'
<DATE> [id]   Python trace: /usr/share/zarafa-dagent/python/mapiplugin.py(13) __init__
<DATE> [id]   Python trace: /usr/share/zarafa-dagent/python/pluginmanager.py(16) loadPlugins
<DATE> [id]   Python trace: /usr/share/zarafa-dagent/python/wraplogger.py(16) logInfo
<DATE> [id] Unable to initialize the dagent plugin manager
```

To fix this issue remove the MAPICore.pyc files from your system. One of the locations can be **/usr/lib/python2.6/dist-packages/MAPICore.pyc**

6.8.6.2. Problem - Solution

No plugins are loaded in the zarafa-dagent

Does the plugin exist in the directory **plugin_path** by default in '/var/lib/zarafa/dagent/plugins'? If not, create a symlink to the plugin to activated or just copy the plugin into the directory.

No plugins are loaded in the zarafa-spooler
Does the plugin exist in the directory **`plugin_path`** by default in '/var/lib/zarafa/spooler/plugins'? If not, create a symlink to the plugin to activated or just copy the plugin into the directory.

6.9. Running ZCP multi-server behind Reverse Proxy

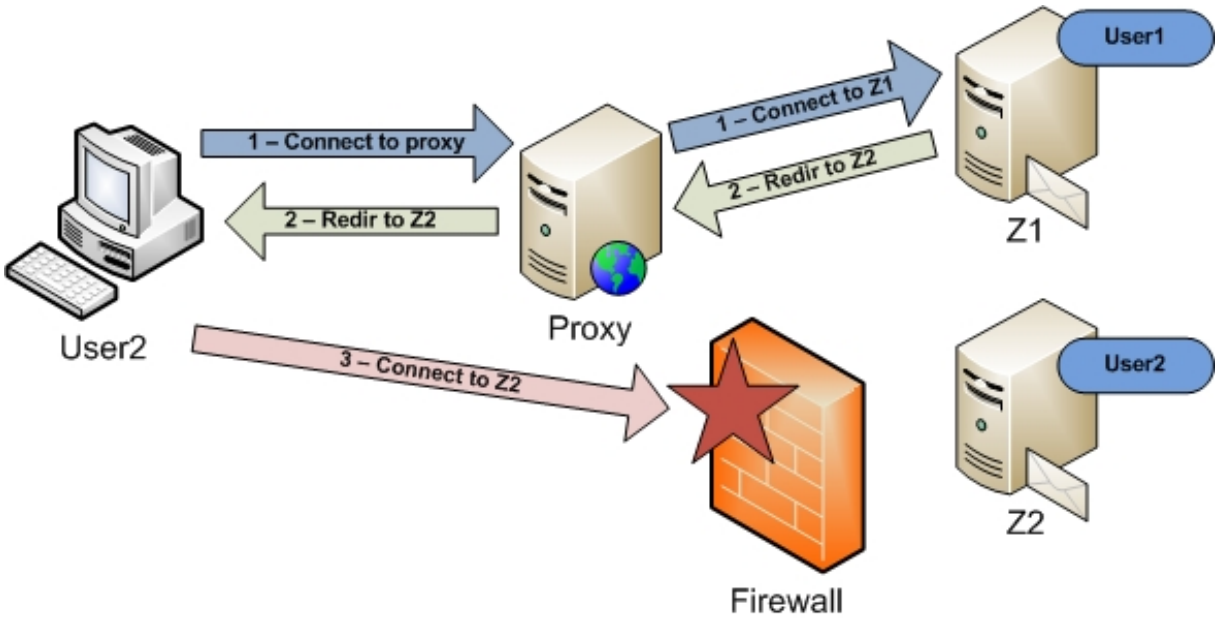
Certain setups require that zarafa-server is not exposed directly to the internet. When offering Outlook access, it is sometimes needed to configure a reverse proxy so that Outlook users can connect to the reverse proxy and not directly to zarafa-server.

Setting up a reverse proxy with a single zarafa-server is quite easy and can be found in chapter 5.1.3 of this administrator manual, however when using a multi-server setup this is a completely different story. Due to the redirection protocol within Zarafa it is quite difficult to setup a reverse proxy for a MutliServer environment, however not impossible.

6.9.1. Description of redirection problem

With redirection the following problem may arise when using a reverse proxy:

1. Outlook connects to a reverse proxy, and the reverse proxy connects to node Z1.
2. Node Z1 will send a redirect for User2 to node Z2.
3. Outlook tries to connect directly to node Z2, but this connection will break on the Firewall.



Therefore zarafa-server has some new options since version 7.1, which will make it easier to setup a reverse proxy for a multi-server environment.

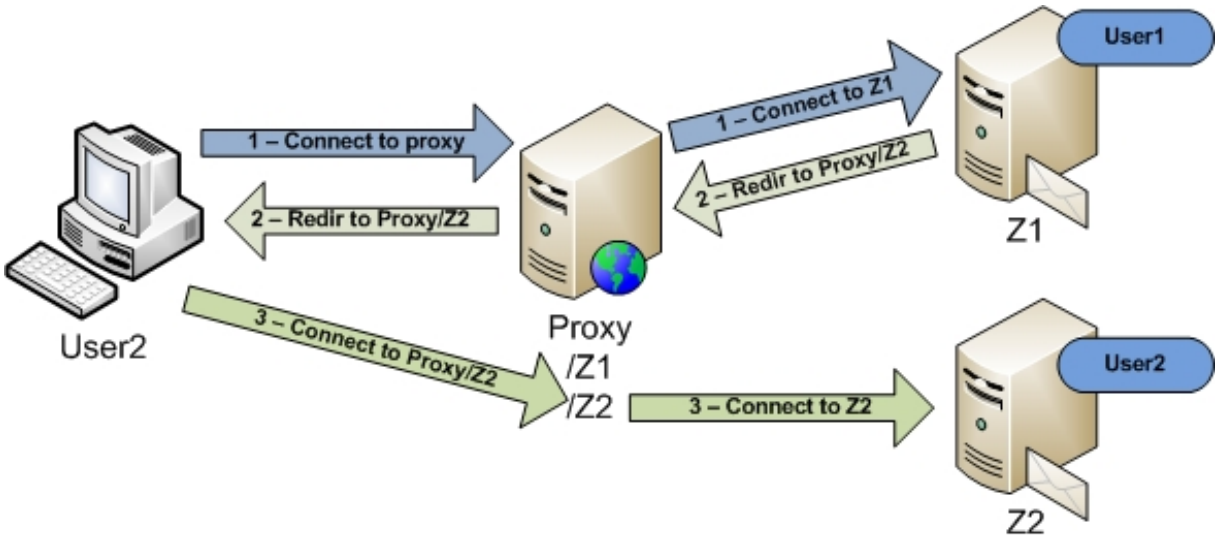
In our new setup the reverse proxy will add extra header information, so the zarafa-server will detect that a connection is being made through a reverse proxy. When a connection is made through a reverse proxy (when the extra header is detected) Zarafa will not reply with the normal redirection string, but it will fetch the connection string from a new **`ldap`** attribute: ZARAFAPROXYURL.

Outlook will then still connect to the reverse proxy, even when a redirect command is given:

1. Outlook connects to the reverse proxy, and the reverse proxy adds the extra header and connects to node Z1.

2. Node Z1 detects the extra header and will send a redirect for User2 to node Proxy/Z2.

3. Outlook will now connect again to the proxy, but with a different path /Z2. The proxy will now connect to Z2 so the store of User2 can be opened.



6.9.2. Setup Prerequisites

When setting up a reverse proxy for a multi-server setup using the new ZCP options, the following prerequisites need to be met:

1. ZCP 7.1 or newer.

2. OpenLDAP or ADS with the schema extensions from ZCP 7.1 or newer.

3. A reverse proxy which fully supports HTTP/1.1 (make sure that also the transport encoding "Chuncked Encoding" is supported).

6.9.3. Example Setup with Apache

Apache 2.2 and newer fully supports HTTP/1.1 in the mod_proxy module.

In our example setup we will use an Apache setup which listens on port 237. In your Apache config you will need to add the following:

```
<IfModule mod_ssl.c>
  NameVirtualHost *:237
  Listen 237
</IfModule>
```

We assume that you have created the correct certificates for Apache, so that Outlook is able to connect using SSL.

6.9.3.1. Configuring Apache

In our example setup we will create a virtual host which is used for reverse proxying:

1. /zarafa will be reverse proxied to node Z1 (Default connection is made to /zarafa)

2. /z1 will be reverse proxied to node Z1 (When a redirection is made to node Z1)

3. /z2 will be reverse proxied to node Z2 (When a redirection is made to node Z2)

In our Apache config we will setup this virtual host:

```
<VirtualHost *:237>
  ServerName zproxy.example.com
  SSLProxyEngine On

  ProxyPass /zarafa https://z1:237/zarafa retry=0
  ProxyPassReverse /zarafa https://z1:237/zarafa retry=0

  ProxyPass /z1 https://z1:237/z1 retry=0
  ProxyPassReverse /z1 https://z1:237/z1 retry=0

  ProxyPass /z2 https://z2:237/z2 retry=0
  ProxyPassReverse /z2 https://z2:237/z2 retry=0

  Header add zarafa_proxy "yes"
  RequestHeader set zarafa_proxy "yes"

  SSLEngine On
  SSLVerifyDepth 2

  SSLCertificateFile /path/to/WEB.CRT
  SSLCertificateKeyFile /path/to/WEB.KEY
  SSLCACertificateFile /path/to/CA.CRT

  CustomLog /var/log/apache2/zproxy.example.com-access.log combined
  ErrorLog /var/log/apache2/zproxy.example.com-error.log
</VirtualHost>
```

Note

When using Apache as a reverse proxy, it is advised to use Apache in a threaded model and not in a prefork model, as the threaded model is able to handle far more concurrent connections then the prefork model.

6.9.3.2. Adding attribute to Servers

We assume you have installed the ZCP 7.1 or newer schema extensions.

In **ldap** add the attribute ZARAFAPROXYURL to all servers in the multi-server environment.

For node Z1 this will be:

```
ZARAFAPROXYURL: https://zproxy.example.com:237/z1
```

So the complete **ldap** record for node Z1 may look something like this:

```
objectClass: top
objectClass: zarafa-server
objectClass: device
objectClass: ipHost
ZARAFAHTTPPORT: 236
ZARAFASSLPORT: 237
ZARAFAFILEPATH: /var/run/zarafa
ipHostNumber: 192.168.1.1
cn: z1
ZARAFAPROXYURL: https://zproxy.example.com:237/z1
```

For node Z2 this will be:

```
ZARAFAPROXYURL: https://zproxy.example.com:237/z2
```

So the complete **ldap** record for node Z2 may look something like this:

```
objectClass: top
objectClass: zarafa-server
objectClass: device
objectClass: ipHost
ZARAFAHTTPPORT: 236
ZARAFASSLPORT: 237
ZARAFAFILEPATH: /var/run/zarafa
ipHostNumber: 192.168.1.2
cn: z2
ZARAFAPROXYURL: https://zproxy.example.com:237/z2
```

6.9.3.3. Configuring Zarafa Server

Now zarafa-server needs to be configured, so that it will send the correct redirect command when the proxy header is detected.

In this example we configured Apache to add the header "zarafa_proxy", if a connection is being made through our reverse proxy.

On all the zarafa servers in the multi-server environment we will need to add an extra config option to the server.cfg:

```
proxy_header = zarafa_proxy
```

Zarafa-server will now send the ZARAFAPROXYURL as redirect string to the client when the header "zarafa_proxy" is detected.

However, internal (´behind´ the proxy) redirections must **not** be redirected to the proxy since this is not necessary. So any internal service (e.g. BES server) will not connect to the reverse proxy, so the extra header is not added and zarafa-server will send the normal redirect string which is generated from the **ldap** database.

The proxy_header option can have different values:

1. Empty: proxy_header option will not be used.
2. [header]: zarafa-server will check for [header], when found zarafa-server send the ZARAFAPROXYURL as redirect string.
3. *: will force zarafa-server to send the ZARAFAPROXYURL as a redirect string everytime a redirect command is given. With this value set, you do not need to add the extra header in your reverse proxy. However also internal (´behind´ the proxy) services will be redirected to the reverse proxy.

Gestion des services ZCP

7.1. Démarrage des services

Il y a 7 services qui peuvent être lancés :

- **zarafa-server**, le processus serveur
- **zarafa-spooler**, envoie le courrier sortant à un serveur SMTP
- **zarafa-monitor**, contrôle le respect des quota
- **zarafa-gateway**, fournit les accès POP3 et IMAP
- **zarafa-ical**, fournit les accès iCal et CALDAV aux clients qui utilisent ce type d'agenda
- **zarafa-licensed**, nécessaire pour utiliser tout module propriétaire Zarafa avec zarafa-server
- **zarafa-search**, fournit un service d'indexation de texte intégral permettant d'effectuer des recherches rapides dans les courriels ou les pièces jointes
- **zarafa-dagent**, exécuté comme service lors de l'utilisation du protocole de transfert de courrier local (LMTP, voir *Section 5.4, « ZCP Postfix integration »*)

Les processus **zarafa-server** et **zarafa-spooler**sont obligatoires pour faire fonctionner Zarafa. Les services **zarafa-monitor**, **zarafa-gateway**, et **zarafa-ical** sont optionnels. Pour démarrer un service, veuillez exécuter :

```
/etc/init.d/zarafa-<nom_du_service> start
```

Remplacer **<nom_du_service>** avec le nom du service qui doit être lancé. Pour lancer **zarafa-server**, saisir :

```
/etc/init.d/zarafa-server start
```

Le script lancera le serveur. Les scripts **init.d** permettent de démarrer (start), arrêter (stop) ou redémarrer (restart) les services. Si le script **init.d** ne peuvent pas être utilisé, le serveur devra être lancé manuellement. Il est possible de préciser explicitement au serveur Zarafa l'emplacement du fichier de configuration, à l'aide de l'option **-c** :

```
/usr/bin/zarafa-server -c /etc/zarafa/server.cfg
```

Le service **zarafa-server** sera lancé en daemon et l'invite de commande réapparaitra presque immédiatement. Utiliser l'option **-F** pour le démarrer au premier plan. L'option **-F** peut également être utilisée pour les programmes, comme daemontools, qui surveillent les services.

7.1.1. Arrêt des services

Pour arrêter un service, veuillez exécuter :

```
/etc/init.d/zarafa-<nom_du_service> stop
```

La plupart des services s'arrêteront immédiatement. Le service **zarafa-spooler** peut prendre environ 10 secondes avant de s'arrêter. Le service **zarafa-server**peut prendre environ 60 secondes avant de s'arrêter.

7.1.2. Rechargement d'une configuration de service

Certaines options peuvent être modifiées et rechargées par le service en cours de fonctionnement. Les options pouvant être rechargées sont décrites dans la page 'man' du fichier de configuration du service. Exemple : pour **zarafa-server**, saisir la commande suivante pour consulter la page 'man' de configuration :

```
man zarafa-server.cfg
```

Sous le chapitre **reloading** se trouvent toutes les options pouvant être rechargées pour ce service. Pour effectuer le rechargement du fichier de configuration, saisir :

```
/etc/init.d/zarafa-<nom_du_service> reload
```

7.2. Options de journalisation

Chaque composant permet à la méthode de journalisation d'être définie dans son fichier de configuration. Les deux méthodes de configuration possibles sont : fichier ou syslog.

Habituellement, la journalisation de chaque composant s'effectue dans un fichier respectif situé dans **/var/log/zarafa**. Ce dossier est créé lors de l'installation des packages. Si ce dossier n'est pas présent ou n'est pas accessible en écriture pour l'utilisateur actuel, les services ne seront pas en mesure d'ouvrir leur fichier de journalisation et imprimeront alors leurs messages sur la sortie standard.

Les messages provenant du fichier de journalisation du serveur peuvent être paramétrés. Les options suivantes doivent être modifiées dans le fichier de configuration :

```
log_method
```

La méthode de consignation des messages. **file** consigne les messages dans un fichier. Sur les systèmes Linux, **syslog** consigne les messages dans le fichier de journalisation du courrier par défaut à l'aide de syslog.

```
log_file
```

Lorsque l'option **log_method** est définie à l'aide de la valeur **file**, c'est cette variable qui configure le nom du fichier. Le serveur doit posséder les permissions d'écriture sur ce répertoire et sur ce fichier.

```
log_level
```

Accroît le niveau des messages qui seront journalisés. Le niveau **6** est le niveau maximum.

```
log_timestamp
```

1 or **0**; Ce paramètre active ou désactive l'horodatage, lors de l'utilisation d'un fichier comme méthode de journalisation.

La journalisation d'autres services que **zarafa-server** se configure de la même façon.

7.3. Journalisation de sécurité

Dans les versions 7.0 et 6.40.7 de Zcp, une fonctionnalité de journalisation supplémentaire de sécurité a été ajoutée. Des auditions basés sur cette journalisation peuvent être effectuées sur le serveur

Zarafa. Cette journalisation contient les messages de démarrage, les authentifications des utilisateurs et les accès aux bases de stockage des délégués.

7.3.1. Éléments de journalisation

7.3.1.1. Démarrage

Lorsque le serveur est (re)démarré, le message suivant sera consigné dans le fichier journal de sécurité :

```
zarafa-server startup by user uid=0 (zarafa-server démarré par l'utilisateur uid=0)
```

L'indicateur suivant peut être présent dans la section démarrage :

uid
: L'ID de l'utilisateur Unix utilisée pour démarrer le serveur (pas nécessairement l'utilisateur employé par le serveur en cours d'exécution)

7.3.1.2. Signaux

Lorsque le serveur reçoit un signal, le message suivant sera consigné dans le journal de sécurité :

```
zarafa-server signalled sig=15 (zarafa-server a reçu le signal sig=15)
```

L'indicateur suivant peut être présent dans la section signal :

sig
: Le signal reçu par le serveur. Consulter **`man 7 signal`** pour une liste des IDs les plus communes de signaux.

7.3.1.3. Authentifications

Lorsqu'un utilisateur (sauf l'utilisateur interne SYSTEM) s'identifie, le message suivant sera consigné dans le journal de sécurité :

Correct authentication: (authentification correcte :)

```
authenticate ok user='john' from='127.0.0.1' method='User supplied password'
 program='apache2'
```

Incorrect authentication: (authentification incorrecte :)

```
authenticate failed user='john' from='127.0.0.1' program='apache2'
```

Uniquement avec des identifiants sso :

```
authenticate spoofed user='john' requested='test' from='192.168.50.178'
 method='kerberos sso' program='OUTLOOK.EXE'
```

Les indicateurs suivants peut être présents dans la section authentification :

user
: Le nom d'utilisateur envoyé au serveur Zarafa.

requested
: Le nom du profil MAPI permettant d'ouvrir la base de stockage, l'indicateur 'user' sera l'utilisateur authentifié. (SSO seulement)

from
: Socket Unix ou adresse IP avec laquelle la connexion au serveur a été effectuée.

method
: La méthode avec laquelle l'utilisateur a été authentifié : socket, certificat, mot de passe, ntlm sso ou kerberos sso.

program
: Le programme utilisé pour l'identification.

7.3.1.4. Authentications with impersonation

When a user logs in and authenticates as another user, the following message will be printed in the security log:

Correct impersonation:

```
authenticate ok user='john' from='127.0.0.1' method='User supplied password'
 program='apache2'
impersonate ok user='jane', from='127.0.0.1' program='apache2'
 impersonator='john'
```

Incorrect impersonation:

```
authenticate ok user='john' from='127.0.0.1' method='User supplied password'
 program='apache2'
impersonate failed user='jane', from='127.0.0.1' program='apache2'
 impersonator='john'
```

The following tags are possible in the impersonation line:

user
: The username of the user being impersonated.

from
: Socket Unix ou adresse IP avec laquelle la connexion au serveur a été effectuée.

program
: Le programme utilisé pour l'identification.

impersonator
: The user that is impersonating another user. This is the user whose credentials are being checked.

7.3.1.5. Sharing actions

Lorsqu'un utilisateur accède à des objets qui ne font pas partie de sa propre base des stockage, un message sera consigné dans le journal. Cela signifie également que des messages seront consignés dans le journal lorsqu'un utilisateur accède à la **`base de stockage publique`**.

Les messages suivants seront consignés dans le journal de sécurité :

Allowed sharing action: (Action partagée permise :)

```
access allowed objectid=387538 type=3 ownername='test' username='constant' rights='view'
```

Denied sharing action: (Action partagée refusée :)

```
access denied objectid=387538 type=3 ownername='test' username='constant' rights='view'
```

Les indicateurs suivants peut être présents dans la ligne de partage :

objectid
: L'objet sur lequel une action a été effectuée

type
: Le type MAPI de l'objet (uniquement base de stockage, dossier ou message).

ownername
: Le propriétaire de la base de stockage qui contient l'objet (pas nécessairement le créateur de l'objet)

username
: L'utilisateur ayant effectué un action sur l'objet.

rights
: Le type d'action effectuée

Note

Dans un environnement de tenant unique, le propriétaire de la **`base de stockage publique`** sera SYSTEM, tandis que dans un environnement multi-tenant ce sera le nom du tenant.

Actions possibles :

read
: Accès d'un objet en lecture

create
: Création d'un nouvel objet

edit
: Modification d'un objet existant (p. ex. modification des propriétés, mais également ajout/retrait de destinataires et de pièces jointes)

delete
: Suppression (logique) ou déplacement de l'objet

create folder
: Création d'un nouveau répertoire

view
: Accès en lecture de l'arborescence des dossiers / des contenus des tableaux

folder permissions
: Modification des permissions, modification ou suppression des répertoires

owner
: Les actions soumettre un message / terminer un message / avorter une soumission / envoi de courriel dans la base de stockage de quelqu'un d'autre ne sont jamais autorisées sauf si vous en êtes propriétaire.

admin
: Inutilisé, ne sera jamais consigné

7.3.1.6. Filtrage du fichier de journalisation

Lorsqu'un utilisateur accède à la base de stockage ou au répertoire d'un délégué, une note est consignée dans le ficher audit.log. Pour obtenir un affichage plus convivial des répertoires des délégués ayant été accédés, le fichier audit.log peut être filtré.

La commande suivante filtrera le fichier de journalisation et rendra son affichage plus convivial :

```
perl /usr/share/doc/zarafa/audit-parse.pl < /var/log/zarafa/audit.log
```

Le script affiche désormais le nom exact du dossier qui a été accédé dans la base de stockage du délégué :

```
access allowed rights='view' type='folder' objectid='store\27\IPM_SUBTREE\Calendar'
 username='john' ownername='mary'
```

Dans cet exemple, l'utilisateur john a consulté l'agenda de l'utilisateur mary.

7.3.1.7. Non consigné dans le journal

Seul les droits des objets de "niveau supérieur" sont contrôlés, de sorte que les actions effectuées sur les pièces jointes, les destinataires, ou les objets incrustés n'apparaitront pas dans le journal.

7.3.2. Configuration

Les options de configuration suivantes ont été ajoutées à **`/etc/zarafa/server.cfg`** :

```
audit_log_enabled = no
audit_log_method = syslog
audit_log_file = -
audit_log_level = 1
audit_log_timestamp = 0
```

Par défaut, la journalisation audit est désactivée. Lorsque elle est activée, les messages seront consignés par défaut à l'aide de syslog car cette méthode peut être configurée pour envoyer des messages à un serveur syslog externe. La valeur **`authpriv`** de l'option facility de syslog sera utilisée pour envoyer les messages.

7.4. Vérification des statistiques de Zarafa

Les statistiques et le statut du serveur peuvent être vérifiés à l'aide de l'utilitaire zarafa-stats. L'utilitaire zarafa-stats offre les options suivantes :

- **`--system`** Donne des informations sur les threads, SQL et les caches
- **`--session`** Donne des informations sur les sessions et le temps pris par le serveur en appels SOAP

- **`--users`** Donne des informations sur les utilisateurs, la taille des bases de stockage et des quotas
- **`--company`** Donne des informations sur les tenants, la place utilisée par les tenants et leur taille des quotas
- **`--top`** Affiche des informations, selon un format similaire à top, sur les sessions et l'utilisation des ressources du serveur

Pour utiliser l'utilitaire zarafa-stats, saisir par exemple la commande suivante :

```
zarafa-stats --top
Last update: Tue Mar 29 13:40:18 2011
Sess: 1     Sess grp: 1       Users: 1    Hosts: 1    CPU: 0%   QLen:    QAge:
SQL/s SEL:    0 UPD:   0 INS:   0 DEL:   0        Threads(idle): ()
        SOAP calls: 6

VERSION      USERID     IP/PID          APP            TIME    CPUTIME CPU    NREQ   TASK
7,0,0,24874  SYSTEM     4527            zarafa-spooler 0:00    0:00    0      6
tableQueryRows
```

L'affichage **`--top`** donne chaque seconde des informations sur le statut de l'utilisation CPU, des clients connectés, des threads actifs, de la longueur des files d'attente et des requêtes SQL. Lorsque le serveur dénote une file d'attente importante et durable, généralement le nombre des threads devrait alors augmenter.

7.5. Système de suppression logique

Lorsqu'un utilisateur supprime un courrier électronique, un élément d'agenda ou un dossier complet, ils sont déplacés par défaut vers le répertoire **`Deleted Items`**.

Lorsque des éléments sont supprimés du répertoire **`Deleted Items`**, ils ne seront pas pour autant supprimés de la base de données. En réalité, ils seront simplement marqués comme étant supprimés afin que l'utilisateur ne puisse plus les visualiser. Même si un utilisateur supprime des éléments à l'aide des touches de raccourci <Maj> <Suppr> ils ne seront toujours pas supprimés de la base de données mais simplement marqués comme étant supprimés.

Ceci simplifie grandement la restauration d'éléments à partir de Microsoft Outlook : sélectionner *Outils* dans la barre de menu Outlook, et cliquer sur *Récupérer les éléments supprimés*. Les éléments sont regroupés selon les dossiers depuis lesquels ils avaient été supprimés. La plupart des éléments apparaîtra dans le dossier *Deleted Items* du fait qu'ils ont été supprimés depuis cette destination.

Les suppressions logiques resteront toujours dans la base de données à moins qu'elles n'en soient purgées. La date de purge d'un élément dépend de la valeur qui a été définie sur l'option **`softdelete_lifetime`**. La valeur par défaut est **`30`** (jours).

Dans cet exemple, la valeur est définie sur **`30`**. Cela signifie que les éléments supprimés seront purgés de la base de données **`30`** jours après leur suppression. Si cette option est définie sur **`0`** (zero), les éléments ne seront jamais retirés de la base de données.

Des purges peuvent également être initiées à l'aide de la commande suivante :

```
zarafa-admin --purge-softdelete <jours>
```

`<jours>` représentant le nombre de jours que les éléments récemment supprimés sont conservés. Si la valeur **`0`** (zero) est définie, tous les éléments supprimés seront purgés.

Pour des raisons de performance il est recommandé de recourir à une purge manuelle du système de suppression logique dans les environnements ZCP de taille importante. Ceci peut être configuré simplement à l'aide d'une tâche Cron.

Gestion des utilisateurs

8.1. Dossier public

Une fois que le serveur a été démarré correctement, les bases de stockage peuvent être créées. Il y a deux types de bases de stockage : les bases privées et les bases publiques. Il ne peut y avoir qu'une seule base publique. Elle peut être créée à l'aide de la commande suivante :

```
/usr/bin/zarafa-admin -s
```

La base de stockage publique est celle qui est accessible à tous les utilisateurs. Après l'installation et la configuration du serveur, une base de stockage publique doit être créée avant que les bases de stockage privées ne puissent être créées à leur tour. Si ZCP est configuré en multi-tenant, une base de stockage publique sera automatiquement créée pour chaque société.

Dans une architecture multi-serveur, la base de stockage publique ne peut être créée que sur le nœud multi-serveur sur lequel l'attribut ZarafaContainsPublic a été activé. Pour le moment, la base de stockage publique ne peut être créée que sur un seul serveur. Consulter *Section 6.3.2, « Préparation / configuration du serveur LDAP dans un environnement multi-serveur »* pour plus d'information.

Note

Par défaut, la base de stockage publique est accessible en lecture et en écriture à tous les utilisateurs. Veuillez réviser les autorisations avant de démarrer le système Zarafa.

8.2. Principales commandes de l'utilitaire zarafa-admin

ZCP offre l'utilitaire d'administration **`zarafa-admin`** pour permettre la gestion des utilisateurs et des groupes Lorsque le plugin **`DB`** est utilisé, l'utilitaire est employé pour créer ou supprimer les utilisateurs et les groupes. Si le plugin **`unix`** ou bien **`ldap`** est utilisé, l'utilitaire ne peut pas être employé pour la création des utilisateurs et des groupes, mais il peut cependant fournir nombre d'information à leur sujet.

Les utilisateurs ou les groupes peuvent tous être affichés à l'aide des commandes suivantes :

```
zarafa-admin -l
zarafa-admin -L
```

Pour afficher plus d'information à propos d'un utilisateur spécifique, saisir la commande suivante :

```
zarafa-admin --details john
Username:              john
Fullname:              John Doe
Emailaddress:          j.doe@example.com
Active:                yes
Administrator:         no
Address book:          Visible
Last logon:            03/25/11 19:50:29
Last logoff:           03/25/11 19:50:29
Quota overrides:       no
Warning level:         1024 MB
Soft level:            2048 MB
Hard level:            3072 MB
```

```
Current store size:    462 MB
Groups (1):
       Everyone
       Sales team
```

Pour afficher plus d'information à propos d'un groupe spécifique, saisir la commande suivante :

```
zarafa-admin --details sales --type group
Groupname:             sales
Fullname:              sales
Emailaddress:
Address book:          Visible
Users (1):
       Username        Fullname        Homeserver
       ------------------------------------------------
       john            John Doe
       mary            Mary Jones
```

Lorsqu'un utilisateur est supprimé, sa boîte aux lettres sera néanmoins conservée dans la base de données. Saisir la commande suivante pour récupérer la liste des bases de stockage sans utilisateur ou les utilisateurs sans base de stockage :

```
/usr/bin/zarafa-admin --list-orphans
Stores without users:
       Store guid                              Guessed username        Last modified
Store size

-------------------------------------------------------------------------------------------------
       CAC27E6D70BB45B0B712B760AE6BA0A8        steve                   2010/03/22 14:22
2334KB

Users without stores:
       Username
       ----------------------------
       jane
```

Il peut être décidé de retirer la base de stockage de la base de données, ou bien d'attacher la base à un autre utilisateur afin de pouvoir de nouveau y accéder. Pour retirer la base de stockage de la base de données, une action qui est irréversible, veuillez saisir la commande suivante :

```
/usr/bin/zarafa-admin --remove-store <store-guid>
```

Pour attacher la base à un autre utilisateur, veuillez saisir la commande suivante :

```
/usr/bin/zarafa-admin --hook-store <store-guid> -u <user>
```

L'utilisateur définit par l'option **-u** pourra alors disposer de la base de stockage. Il faut se reconnecter sur WebAccess ou créer un nouveau profil dans Outlook afin d'accéder à la base.

Important

Lorsqu'une base de stockage est ainsi attachée à un utilisateur qui possédait déjà une autre base de stockage, celle-ci deviendra alors orpheline. La base de stockage d'origine pourra ensuite être retrouvée à l'aide de l'option **list-orphans** de la commande **zarafa-admin**.

Note

Dans ZCP 6.30.6 et dans les versions précédentes, la base d'un l'utilisateur était déplacée dans le dossier “Deleted Stores” du dossier public, à la suite de la suppression de son propriétaire. Ce dossier n'est accessible qu'aux administrateurs. Les administrateurs peuvent ainsi parcourir les dossiers ou bien purger définitivement les bases supprimées en supprimant leur dossier correspond du répertoire “Deleted stores”. Ceci s'applique quel que soit le plugin utilisateur utilisé.

Les informations complètes sur toutes les options de **`zarafa-admin`** sont disponibles dans la page man de l'utilitaire.

```
man zarafa-admin
```

8.3. Gestion des utilisateurs avec le plugin DB

Par défaut le plugin DB sera utilisé pour la gestion des utilisateurs. Les commandes permettant de gérer les utilisateurs à l'aide de **`zarafa-admin`** sont décrite ci-dessous. Pour l'administration des utilisateurs à l'aide du plugin LDAP, veuillez consulter *Section 8.5, « Gestion des utilisateurs avec LDAP ou Active Directory »*.

Pour le moment ZCP ne fournit pas d'interface graphique d'administration, cependant plusieurs applications tierces proposent des consoles d'administration du système Zarafa en interface Web.

8.3.1. Création des utilisateurs avec le plugin DB

Pour créer un nouvel utilisateur, veuillez saisir la commande suivante :

```
/usr/bin/zarafa-admin -c <user name> -p <password> \
       -e <email> -f <full name> -a <administrator>
```

Les champs entre <> doivent être remplis de la façon suivante :

- **User name:** Le nom de l'utilisateur. C'est avec cet identifiant que l'utilisateur se connectera à la base.
- **Password:** Le mot de passe en clair. Le mot de passe sera chiffré et stocké dans la base de données.
- **Email:** L'adresse électronique de l'utilisateur. Généralement c'est **`<user name>@<email domain>`**.
- **Full name:** Le nom complet de l'utilisateur. Comme le nom complet comprend des espaces et éventuellement des caractères non alphanumériques, il doit être saisi entre deux guillemets (**`''`**).
- **Administrator:** cette valeur doit être **`0`** ou **`1`**. Un utilisateur qui obtient la qualité d'administrateur sera en mesure d'accéder à toutes les bases de stockage de n'importe quel autre utilisateur. Il est également possible d'utiliser la valeur **`2`** comme niveau d'administrateur, auquel cas, l'administrateur pourra également accéder aux bases de stockage des utilisateurs des autres entreprises.

Tous les champs, excepté celui de l'adresse électronique, sont sensibles à la casse.

Le mot de passe peut également être renseigné en utilisant l'option **-P**. Dans ce cas, le mot de passe ne sera pas donné à l'invite de commande, mais il sera demandé par l'utilitaire **zarafa-admin**. Le mot de passe ne s'affichera pas sur l'écran et devra être renseigné deux fois de suite pour vérification.

8.3.2. Utilisateurs non-actifs

Un utilisateur non-actif ne peut pas s'authentifier directement sur ZCP, mais son courriel peut être livré et son dossier peut être accessible aux utilisateurs possédant les permissions adéquates. Les utilisateurs non-active sont particulièrement utiles comme boîte aux lettres fonctionnelles ou comme ressources.

Pour créer un utilisateur non-actif, veuillez utiliser la commande suivante :

```
zarafa-admin -c <user name> -P -e <email> -f <full name> -n 1
```

Note

Dans ZCP 6.30 et dans les versions précédentes, il n'est pas possible de basculer un utilisateur actif en non-actif ou vice versa. Le basculement de la valeur non-active entraine la suppression de la boîte aux lettres.

8.3.3. Actualisation des informations des utilisateurs avec le plugin DB

L'utilitaire **zarafa-admin** permet également d'actualiser les informations se rapportant aux bases de stockages et aux utilisateurs. Veuillez saisir la commande suivante pour actualiser les informations des utilisateurs :

```
/usr/bin/zarafa-admin -u <nom de l'utilisateur> [-U <nouveau nom de l'utilisateur>] \
       [-p <nouveau mot de passe>] [-e <adresse électronique>] \
       [-f <nom complet>] [-a <0|1>]
```

Toutes les modifications sont facultatives. Par exemple, seul le mot de passe d'un utilisateur existant peut être actualisé tandis que les autres informations de l'utilisateur resteront inchangées.

8.3.4. Suppression des utilisateurs avec le plugin DB

Pour supprimer un utilisateur du serveur, veuillez saisir la commande suivante :

```
/usr/bin/zarafa-admin -d <user name>
```

L'utilisateur sera supprimé de la base de données. Cependant, sa base de stockage sera préservée dans la base de donnée, sans être accessible pour autant. Consulter *Section 8.2, « Principales commandes de l'utilitaire zarafa-admin »* pour plus d'information sur la gestion des bases de stockage orphelines.

8.3.5. Configuration des permissions ‘Envoyer en tant que’

ZCP gère deux types de délégation d'envoi :

Permissions 'Envoyer au nom de'

Si un utilisateur confère à un autre utilisateur les droits nécessaires, ce dernier peut envoyer des éléments ‘au nom de’ ce premier utilisateur. Dans ce cas, un courrier ou une demande de rendez-vous

sera envoyé avec le champ “from” suivant : **<delegate>** au nom de **<user>**. Ce paramètre ne peut être configuré qu'à partir de WebAccess ou du client Outlook.

Permissions 'Envoyer en tant que'

Si l'administrateur système confère à l'utilisateur B les droits nécessaires pour ‘Envoyer en tant que’ l'utilisateur A, le destinataire du message ne verra pas que c'est l'utilisateur B qui l'a envoyé. Seules les références de l'utilisateur A seront affichées dans le champ “De”.

La mise en œuvre de délégation à l'aide de **zarafa-admin** n'est possible qu'avec le plugin DB ou UNIX. Pour les environnements LDAP ou Active Directory consulter *Section 8.5, « Gestion des utilisateurs avec LDAP ou Active Directory »*.

Ajouter un utilisateur à la liste d'un délégué qui est actualisé avec la nouvelle valeur de la permission ‘Envoyer en tant que’ cet utilisateur. Ce délégué peut alors envoyer des courriers 'en tant que' l'utilisateur actualisé, à moins que l'utilisateur actualisé n'ait lui-même défini ce délégué dans sa propre liste. Cette option n'est valide qu'avec le paramètre d'actualisation **-u**.

```
zarafa-admin -u <delegate> --add-sendas <user>
```

Par exemple :

```
zarafa-admin -u helpdesk --add-sendas john
```

Retire un utilisateur de la liste d'un délégué qui est actualisé avec la nouvelle valeur de la permission ‘Envoyer en tant que’ cet utilisateur. Cette option n'est valide qu'avec le paramètre d'actualisation **-u**.

```
zarafa-admin -u <delegate> --del-sendas <user>
```

Cette commande affiche tous les utilisateurs contenus dans la liste du délégué.

```
zarafa-admin --list-sendas helpdesk
Send-as list (1) for user helpdesk:
       Username        Fullname
       ----------------------------
       john            John Doe
```

Note

Si le plugin DB est utilisé, les permissions 'Envoyer en tant que' ne peuvent pas être définies pour des groupes.

Note

Lorsque les permissions "Envoyer au nom de" et "Envoyer en tant que " sont toutes deux définies sur le même utilisateur, le courrier sera toujours envoyé "au nom de".

8.3.6. Groupes

Le serveur prend les groupes en charge. Les utilisateurs peuvent appartenir à un nombre quelconque de groupes. Tous les utilisateurs appartiennent toujours au groupe spécial 'Everyone'. La configuration des paramètres de sécurité s'effectue de la même façon sur les dossiers et les éléments des utilisateurs que sur ceux des groupes.

Par exemple, le groupe 'Everyone' dispose de l'accès en lecture vers la boîte de réception de Peter. Dans ce cas, tous les utilisateurs pourront accéder au courrier dans la boîte de réception de Peter, car tous les utilisateurs sont membres du groupe 'Everyone'.

Lorsqu'un nouvel utilisateur Zarafa est créé, par défaut seules les informations Libre/Occupé sont accessibles en lecture au groupe 'Everyone'.

8.3.6.1. Création des groupes avec le plugin DB

À l'aide de l'utilitaire **`zarafa-admin`**, il est possible de créer des groupes, et d'y ajouter ou supprimer des utilisateurs. Dans l'exemple suivant, l'utilisateur john est créé, ainsi que le groupe administration, puis l'utilisateur john est ajouté au groupe administration.

```
zarafa-admin -c john -p secret -f “John Doe” -e “j.doe@domain.com”
zarafa-admin -g administration
zarafa-admin -b john -i administration
```

À l'aide des options **`-l`** ou **`-L`**, une liste d'utilisateurs ou de groupes peut être établie à partir du serveur

Tous les utilisateurs créés seront membre du groupe “Everyone”, ceci ne peut pas être modifié. Les groupes créés avec le plugin DB plugin peuvent être utilisés autant pour la gestion des permissions que pour l'envoi de courriel à un groupe spécifique.

8.4. Gestion des utilisateurs avec le plugin UNIX

Si ZCP est intégrée avec les utilisateurs et groupes par défaut du serveur Linux, une partie de l'administration des utilisateurs devra être effectuée à l'aide des utilitaires Linux classiques tels que **`useradd`** tandis que la partie spécifique se rapportant à Zarafa devra être effectuée avec l'utilitaire **`zarafa-admin`** tool.

8.4.1. Création des utilisateurs avec le plugin UNIX

Pour créer un nouvel utilisateur, utiliser la commande **`adduser`**.

```
useradd <username> -c "Full name"
passwd <username>
```

Comme l'adresse électronique d'un utilisateur ne peut être spécifiée avec la commande **`adduser`**, l'adresse électronique par défaut sera <username>@default_domain. Le domaine pare défaut est spécifié dans le fichier de configuration **`/etc/zarafa/unix.cfg`**.

Cette adresse électronique pourra ensuite être modifiée à l'aide de l'utilitaire **`zarafa-admin`**.

```
zarafa-admin -u <username> -e <email address>
```

8.4.2. Utilisateurs non-actifs

Un utilisateur non-actif ne peut pas s'authentifier directement sur ZCP, mais son courriel peut être livré et son dossier peut être accessible aux utilisateurs possédant les permissions adéquates. Les utilisateurs non-active sont particulièrement utiles comme boîte aux lettres fonctionnelles ou comme ressources.

Pour créer un utilisateur non-actif avec le plugin UNIX, il faut s'assurer que la valeur du shell de l'utilisateur soit définie sur **`/bin/false`**. Le shell pour les utilisateurs non-actifs peut également être définie dans le fichier de configuration **`/etc/zarafa/unix.cfg`**.

Note

Dans ZCP 6.30 et dans les versions précédentes, il n'est pas possible de basculer un utilisateur actif en non-actif ou vice versa. Le basculement de la valeur non-active entraine la suppression de la boîte aux lettres.

8.4.3. Actualisation des informations des utilisateurs avec le plugin UNIX

L'actualisation des informations des utilisateurs avec le plugin UNIX peut s'effectuer pour une part avec les utilitaires Linux classiques et pour l'autre part à l'aide de l'utilitaire **`zarafa-admin`**.

Les informations suivantes devront être modifiées dans le fichier **`/etc/passwd`** ou à l'aide des utilitaires Linux classiques de gestion des utilisateurs :

- Identifiant
- Mot de passe
- Nom complet
- Type de boîte aux lettres (active ou non-active)
- Appartenance à un groupe

Les autres informations suivantes devront être modifiées et configurées à l'aide de l'utilitaire **`zarafa-admin`**.

- Adresse électronique
- Status administrateur
- Quota
- Permissions 'Envoyer en tant que'

8.4.4. Suppression des utilisateurs avec le plugin UNIX

Pour supprimer un utilisateur du serveur, saisir la commande Linux suivante :

```
userdel <username>
```

L'utilisateur sera supprimé de la base de données. Cependant, sa base de stockage sera préservée dans la base de donnée, sans être accessible pour autant. Consulter *Section 8.2, « Principales commandes de l'utilitaire zarafa-admin »* pour plus d'information sur la gestion des bases de stockage orphelines.

8.4.5. Configuration des permissions ‘Envoyer en tant que’

ZCP gère deux types de délégation d'envoi :

Permissions 'Envoyer au nom de'

Si un utilisateur confère à un autre utilisateur les droits nécessaires, ce dernier peut envoyer des éléments ‘au nom de’ ce premier utilisateur. Dans ce cas, un courrier ou une demande de rendez-vous

sera envoyé avec le champ “from” suivant : **<delegate>** au nom de **<user>**. Ce paramètre ne peut être configuré qu'à partir de WebAccess ou du client Outlook.

Permissions 'Envoyer en tant que'

Si l'administrateur système confère à l'utilisateur B les droits nécessaires pour ‘Envoyer en tant que’ l'utilisateur A, le destinataire du message ne verra pas que c'est l'utilisateur B qui l'a envoyé. Seules les références de l'utilisateur A seront affichées dans le champ “De”.

Ajouter un utilisateur à la liste d'un délégué qui est actualisé avec la nouvelle valeur de la permission ‘Envoyer en tant que’ cet utilisateur. Ce délégué peut alors envoyer des courriers 'en tant que' l'utilisateur actualisé, à moins que l'utilisateur actualisé n'ait lui-même défini ce délégué dans sa propre liste. Cette option n'est valide qu'avec le paramètre d'actualisation **-u**.

```
zarafa-admin -u <delegate> --add-sendas <user>
```

Par exemple :

```
zarafa-admin -u helpdesk --add-sendas john
```

Retire un utilisateur de la liste d'un délégué qui est actualisé avec la nouvelle valeur de la permission ‘Envoyer en tant que’ cet utilisateur. Cette option n'est valide qu'avec le paramètre d'actualisation **-u**.

```
zarafa-admin -u <delegate> --del-sendas <user>
```

Cette commande affiche tous les utilisateurs contenus dans la liste du délégué.

```
zarafa-admin --list-sendas helpdesk
Send-as list (1) for user helpdesk:
       Username        Fullname
       ----------------------------
       john            John Doe
```

Note

Si le plugin UNIX est utilisé, les permissions 'Envoyer en tant que' ne peuvent pas être définies pour des groupes.

Note

Lorsque les permissions "Envoyer au nom de" et "Envoyer en tant que " sont toutes deux définies sur le même utilisateur, le courrier sera toujours envoyé "au nom de".

8.4.6. Gestion des groupes avec le plugin UNIX

Le serveur prend les groupes en charge. Les utilisateurs peuvent appartenir à un nombre quelconque de groupes. Tous les utilisateurs appartiennent toujours au groupe spécial 'Everyone'. La configuration des paramètres de sécurité s'effectue de la même façon sur les dossiers et les éléments des utilisateurs que sur ceux des groupes.

Par exemple, le groupe 'Everyone' dispose de l'accès en lecture vers la boîte de réception de Peter. Dans ce cas, tous les utilisateurs pourront accéder au courrier dans la boîte de réception de Peter, car tous les utilisateurs sont membres du groupe 'Everyone'.

Lorsqu'un nouvel utilisateur Zarafa est créé, par défaut seules les informations Libre/Occupé sont accessibles en lecture au groupe 'Everyone'.

8.4.6.1. Création des groupes avec le plugin UNIX

Des groupes peuvent être créés et des utilisateurs peuvent y être ajoutés ou retirés à l'aide des utilitaires Linux classiques de gestion des utilisateurs. Dans l'exemple suivant, le groupe administration est créé, puis l'utilisateur john est ajouté au groupe administration.

```
groupadd administration
usermod -a -G administration john
```

À l'aide des options **`-l`** ou **`-L`**, une liste d'utilisateurs ou de groupes peut être établie à partir du serveur

Tous les utilisateurs créés seront membre du groupe “Everyone”, ceci ne peut pas être modifié. Les groupes créés avec le plugin UNIX plugin peuvent être utilisés autant pour la gestion des permissions que pour l'envoi de courriel à un groupe spécifique.

8.5. Gestion des utilisateurs avec LDAP ou Active Directory

Le serveur Zarafa est un système qui permet à son administrateur de spécifier un serveur de type LDAP pour l'obtention des informations d'utilisateurs, de groupes et de sociétés. Cela signifie que la gestion des utilisateurs peut être simplifiée et que les utilitaires d'administration LDAP habituels peuvent être utilisés. De plus, l'utilisation d'un serveur LDAP permet l'intégration de Zarafa dans un environnement existant.

Les différents systèmes de serveurs LDAP sont reconnus, et Zarafa communiquera avec tout protocole LDAP standard de version 3 ou plus. Cela signifie que Zarafa peut être associé avec d'autres solutions conventionnelles telles que Microsoft Active Directory, OpenLDAP et eDirectory.

Cette section apporte une présentation générale de l'utilisation, par Zarafa, d'un serveur LDAP comme source d'information pour les utilisateurs, les groupes et les sociétés. Dans la plupart des cas, une configuration particulière nécessitera d'autres options et paramétrages supplémentaires que celles qui sont décrites dans ce document. Il est donc présupposé que le lecteur possède une bonne compréhension du fonctionnement des arborescences LDAP et de leur configuration dans le réseau.

Pour plus d'informations, veuillez consulter les exemples de configuration, ainsi que les pages 'man', disponibles sur tous les systèmes sur lesquels Zarafa a été installé.

8.5.1. Les principes de la synchronisation utilisateur de Zarafa

Dans chaque serveur Zarafa , une base de données conservent les informations nécessaires à l'exécution de Zarafa. A part les données se rapportant aux répertoires et leurs divers éléments, la base de données conserve également des informations concernant les droits d'accès aux données, les paramètres des utilisateurs ainsi que des méta-données d'utilisateurs et de groupes. Beaucoup de ces données se réfèrent à l'ID d'un utilisateur spécifique. Par exemple, une ACL (liste de contrôle d'accès) pour la 'boîte de réception' de l'utilisateur A sera stockée dans la base de données en tant qu'enregistrement de la table ACL. Cet enregistrement contient les droits d'accès actuels pour les objets, ainsi que l'ID de l'utilisateur auquel la valeur du contrôle d'accès a été assignée.

L'ID de l'utilisateur évoqué ci-dessus est donc une référence pour l'ID d'un utilisateur à l'intérieur de la base de donnée de Zarafa. L'ID est stockée dans la table 'users', en même temps qu'une référence à l'ID de l'utilisateur dans la base de données des utilisateurs externe (ici un serveur LDAP). Par exemple l'utilisateur ‘A’ peut avoir l'ID utilisateur **`5`** dans le système Zarafa, et peut se référer à l'élément (**`dn=cn=user,dc=example,dc=com`**) sur le serveur LDAP.

Tenir une liste d'utilisateurs de cette façon résout à la fois le problème de la création d'une base de stockage pour un utilisateur, en effet, il est impossible de déclencher la création d'une base de stockage sur le serveur Zarafa à chaque fois qu'un utilisateur est ajouté dans le serveur LDAP. La table ‘users’ fournit une méthode pratique pour détecter les nouveaux utilisateurs qui ont été ajoutés au système et qui nécessitent la création d'une nouvelle base. Il en va de même lors de la suppression d'utilisateurs, puisque les bases d'utilisateurs doivent être effacées en même temps que ceux-ci.

Ainsi, la table ‘users’ est presque exclusivement une correspondance entre l'ID de l'utilisateur qui est utilisé à l'intérieur du système Zarafa, et une référence externe vers un utilisateur d'une base de données LDAP. Bien entendu, pour tout ajout ou suppression d'utilisateurs dans le serveur LDAP, cette table doit être synchronisée avec ces changements.

Plusieurs méthodes permettent d'assurer la synchronisation de la table ‘users’ avec le serveur LDAP, mais Zarafa a opté par défaut pour une approche ‘juste-à-temps’. Cela signifie qu'à chaque fois qu'un utilisateur est requis par le système, son existence est premièrement vérifiée dans le serveur LDAP, puis dans la table ‘users’. Si l'utilisateur n'existe pas sur le serveur Zarafa, il est alors créé instantanément, avant que l'information ne soit renvoyée à l'instigateur de la requête.

La synchronisation semblera instantanée aux utilisateurs et aux administrateurs ; il n'y aura aucun délai entre l'ajout et la suppression d'utilisateurs depuis le serveur LDAP et leur apparition ou disparition dans Zarafa.

Puisque tous les composants de Zarafa utilisent la même interface MAPI pour se connecter à l'infrastructure du serveur, aucun problème lié à la synchronisation de la base de donnée utilisateur ne peut se produire avec l'un des utilitaires de Zarafa. Par exemple la distribution d'un courriel à un utilisateur qui vient juste d'être créé ne pourra jamais échouer du fait que cet utilisateur n'existerait pas dans la table d'utilisateurs de Zarafa.

Un paramètre optionnel *sync_gab_realtime* est disponible dans le fichier de configuration **`server.cfg`** pour optimiser cette synchronisation avec des carnets d'adresses globaux très volumineux de LDAP. Si cette option est définie sur *no* il n'y a aucune synchronisation en temps réel entre l'annuaire LDAP et le serveur Zarafa. Dans ce cas, toutes les données du carnet d'adresses global seront obtenues à partir du cache du serveur Zarafa. Ceci est particulièrement utile dans les infrastructures qui possèdent des carnets d'adresses très volumineux (plus de 10000 enregistrements dans le carnet d'adresses).

La synchronisation entre l'annuaire LDAP et le serveur Zarafa doit être forcée à l'aide de la commande suivante :

```
zarafa-admin --sync
```

Cette commande peut être exécutée un certain nombre de fois par jour ou par heures dans un cronjob.

8.5.1.1. Ajout/Suppression d'évènements

Le mécanisme ci-dessus crée une situation dans laquelle six évènements peuvent être signalés :

- Création d'utilisateur
- Création de groupe
- Création de société
- Suppression d'utilisateur

- Suppression de groupe
- Suppression de société

Ces six évènements peuvent être associés à un script (qui sera décrit plus tard) pour permettre aux administrateurs du système d'accomplir certaines actions spécifiques sur leurs serveurs en même temps que ces évènements. Par défaut, Zarafa n'exécutera que les actions absolument nécessaires pendant ces évènements, par exemple, la création ou la suppression de bases de stockage. Tout autre évènement peut être mis en script par l'administrateur du système. Cela signifie que par défaut, aucune action ne sera exécutée pendant la création ou la suppression de groupes.

8.5.1.2. Appartenance à un groupe

Zarafa synchronise les utilisateurs, les groupes et les sociétés de façon à leur attribuer une ID d'utilisateur, cependant l'information du groupe des utilisateurs n'est jamais stockée sur le serveur Zarafa. Cela signifie que les changements d'appartenance à un groupe se font eux aussi en temps-réel, et que le serveur Zarafa vérifiera l'appartenance à un groupe pour un utilisateur ou pour une liste d'utilisateurs directement à partir du serveur LDAP. La façon dont la correspondance entre les membres d'un groupe et les utilisateurs est effectuée sera abordée plus tard.

8.5.1.3. Dépendance du serveur LDAP

Comme la base de donnée ‘users’ de Zarafa ne détient pas réellement l'information relative aux utilisateurs ou aux groupes, mais seulement une référence vers le serveur LDAP, le serveur Zarafa ne peut fonctionner sans un serveur LDAP qui soit accessible et en fonctionnement. Si le serveur LDAP est défaillant pendant le fonctionnement de Zarafa, les utilitaires Zarafa ne pourront exécuter aucune action, étant donné que la plupart des actions côté serveur exigent un type d'interaction avec le serveur LDAP. Par exemple, le simple fait d'ouvrir un courrier électronique suppose une interrogation au serveur LDAP afin de connaître les groupes auxquels l'utilisateur actuel appartient.. Ce n'est qu'après avoir obtenu cette information que Zarafa peut déterminer si l'utilisateur actuel possède les permissions nécessaires lui permettant d'ouvrir le message.

Si OpenLDAP est employé comme source LDAP, il est recommandé d'utiliser une duplication LDAP pour garantir que le serveur LDAP soit utilisable en toute circonstance en exécutant un serveur OpenLDAP sur la même machine que Zarafa. Ceci assurera que le serveur local LDAP soit toujours joignable, et que Zarafa puisse toujours continuer à fonctionner normalement.

8.5.1.4. Paramétrer l'annuaire LDAP

Bien qu'en principe la plupart des annuaires LDAP peuvent être utilisés avec Zarafa, cette section décrit la façon dont Zarafa effectue ses requêtes de données au serveur LDAP, et comment ces données sont utilisées par le serveur Zarafa et par ses utilitaires.

Les informations suivantes sont requises du serveur LDAP :

- Détails sur l'utilisateur (nom, adresse électronique, etc)
- Contacts (nom, adresse électronique)
- Détails sur le groupe (nom du groupe)
- Détails sur la société
- Relation Utilisateur/Groupe (appartenance de groupe)
- Membres de la société (appartenance d'utilisateur et de groupe)

- Relation Société (affichage transversal des sociétés et permissions d'administration)

Il est possible de paramétrer les objets classés comme utilisateur, contact, groupe, groupe dynamique, liste d'adresses ou société ainsi que les attributs de données à l'aide des fichiers de configuration de Zarafa afin que Zarafa puisse répondre aux demandes du schéma LDAP. Cependant, certaines règles permettront de préserver la clarté de l'annuaire LDAP et en faciliteront sa gestion :

- Toujours utiliser une interface graphique pour la gestion des utilisateurs ou des groupes. De nombreux utilitaires de gestion LDAP existent. (Par exemple, *phpLDAPadmin*[1] pour OpenLDAP en interface Web)
- Si certains utilisateurs utilisent Zarafa et d'autres non, il est préférables de regrouper ces utilisateurs dans des ‘dossiers’ séparés. Un enregistrement **`OU`** ou tout autre objet **`dc-type`** peuvent être employé pour créer ces dossiers.
- Si Microsoft Active Directory est utilisé, il faut s'assurer que les utilisateurs réels soient dans des dossiers LDAP séparés, de façon que Zarafa n'ait pas besoin d'importer des utilisateurs standards comme ‘Administrator’ et ‘Guest’ dans la base de données. Il est également possible de filtrer les utilisateurs à l'aide d'une requête de recherche LDAP, mais les requêtes de recherche peuvent devenir très longues avec ADS.

En règle générale, il faut toujours employer le protocole LDAPS (SSL) lors des connexions au serveur LDAP. Si SSL n'est pas utilisé, l'information sera transmise en clair lors du transfert. Cela ouvre des possibilités de 'reniflage' des mots de passe utilisateur (et administrateur !) lors de leur transfert sur le réseau. Zarafa gère les connexions LDAPS à l'aide de SSL et d'un certificat spécifié dans **`/etc/ldap/ldap.conf`** qui est compatible à la fois avec Microsoft Active Directory et avec les serveurs OpenLDAP. Cependant pour le moment, Zarafa ne gère pas le chiffrage de type **`STARTTLS`**. Pour plus d'information sur la configuration de Active Directory avec prise en charge SSL, veuillez consulter *http://wiki.zarafa.com*.

8.5.2. Gestion des utilisateurs avec ADS

8.5.2.1. Création des utilisateurs avec ADS

Les nouveaux utilisateurs peuvent être créés à l'aide de l'assistant classique de création d'un nouvel utilisateur de Active Directory. Lors de la création d'un utilisateur, il faut toujours s'assurer que l'adresse électronique par défaut de l'utilisateur est unique.

Pour configurer les informations d'utilisateurs spécifiques à Zarafa sélectionner l'onglet **`Zarafa`** de l'utilisateur dans Active Directory.

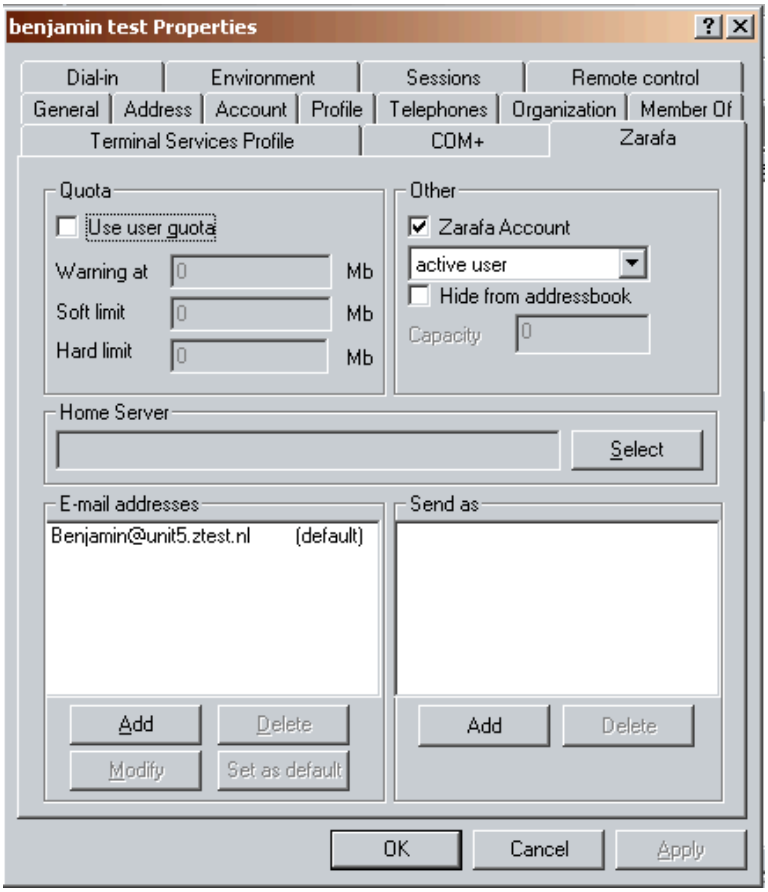


Figure 8.1. Onglet de l'utilisateur Zarafa

8.5.2.2. Création des groupes avec ADS

Des groupes de sécurité peuvent être créés dans Active Directory de même que des groupes de diffusion. Les groupes de sécurité peuvent être utilisés afin de définir les permissions et pour l'envoi de courriel. Les groupes de diffusion ne peuvent être utilisés que pour envoyer du courrier et ne seront **pas** affichés lors de la configuration d'autorisations sur un dossier.

ZCP 6.40 et ses versions supérieures gèrent les groupes imbriqués

Ces groupes peuvent être créés à l'aide de l'assistant classique de création d'un nouvel utilisateur de Active Directory.

8.5.2.3. Création de contacts avec ADS

Le carnet d'adresses global peut recevoir des contacts. Les contacts sont des adresses SMTP externes qui sont affichées dans le carnet d'adresses global et qui peuvent être utilisés comme membres d'une lise de diffusion.

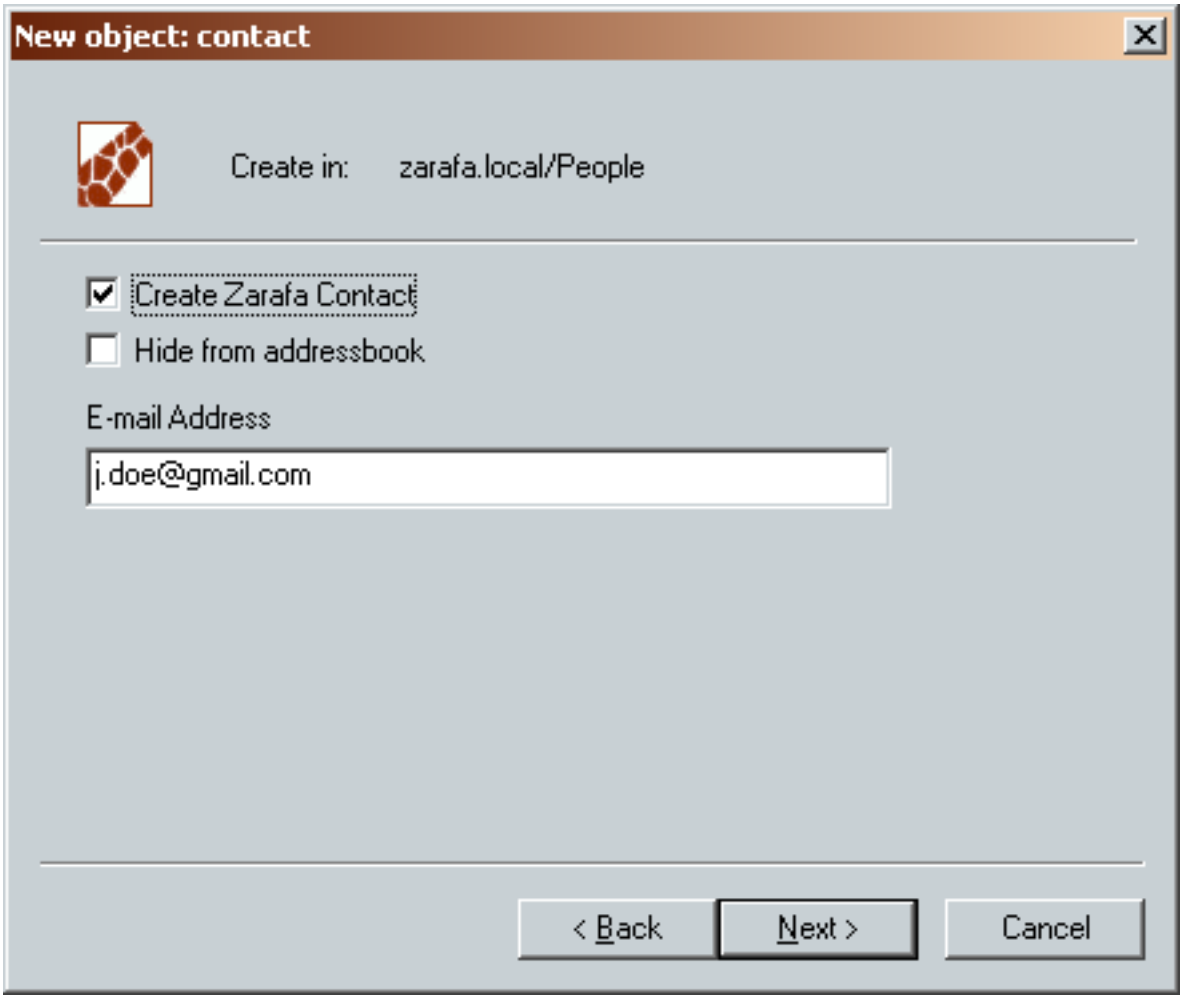


Figure 8.2. Assistant de création de contact

8.5.2.4. Configuration des permissions 'Envoyer en tant que' avec ADS

Les permissions 'Envoyer en tant que' peuvent être définies autant sur les utilisateurs que sur les contacts. Les utilisateurs et les groupes qui aspirent à pouvoir envoyer en tant qu'une adresse spécifique doivent être ajoutés à la liste des privilèges 'Envoyer en tant que' de l'utilisateur ou du contact.

Pour vérifier si les permissions ont été correctement configurées, saisir la commande suivante :

```
zarafa-admin --list-sendas <username>
```

Par exemple :

```
zarafa-admin --list-sendas helpdesk
Send-as list (1) for user helpdesk:
       Username        Fullname
       ----------------------------
       john            John Doe
```

Les utilisateurs possédant les permissions 'Envoyer en tant que', devrait alors pouvoir ajouter l'autre adresse dans les champs‘De’ et ‘Envoyer en tant que’ depuis leur compte.

Depuis ZCP 6.40 le système 'Envoyer en tant que' a évolué :

- La configuration des permissions ‘Envoyer en tant que’ s'effectue d'une façon opposée à celle des versions précédentes de Zarafa. Les permissions ‘Envoyer en tant que’ doivent dorénavant être configurées sur l'utilisateur qui est sélectionné 'En tant que' dans le champ 'De'.
- Consulter *Section 3.5.1, « Étapes de mise à jour pré 6.40 »* pour la conversion des permissions 'Envoyer en tant que'.
- Les groupes peuvent maintenant également être utilisés pour configurer les permissions 'Envoyer en tant que'.

8.5.2.5. Alias d'utilisateur 'Envoyé en tant que'

Sous l'onglet utilisateur de Active Directory, des alias de courriel peuvent être attribués à l'utilisateur. Ces alias ne sont utilisés que pour les courriers entrants. Par défaut l'envoi d'un courriel avec l'alias n'est pas possible, cependant l'astuce suivante peut être utilisée :

1. Créer un nouveau contact dans ADS pour chaque adresse que vous voulez utiliser pour envoyer le courrier sortant
2. Le contact peut être rendu invisible afin de le cacher du Carnet d'Adresses Global
3. S'assurer que l'alias est défini comme l'adresse principale du contact
4. Assigner à l'utilisateur les permissions 'Envoyer en tant que' sur le nouveau contact
5. Lors de l'envoi d'un nouveau message, l'utilisateur devrait alors pouvoir manuellement sélectionner le champ 'De' dans Outlook ou bien ajouter un compte additionnel dans les paramètres de WebAccess.

8.5.2.6. Paramétrer les listes d'adresses dans ADS

Les listes d'adresses sont des sous-ensembles du carnet d'adresses global qui répondent à des critères spécifiques. Par exemple, vous pouvez créer une liste d'adresses qui contient tous les utilisateurs de Manchester et une autre contient tous les utilisateurs de Stuttgart.

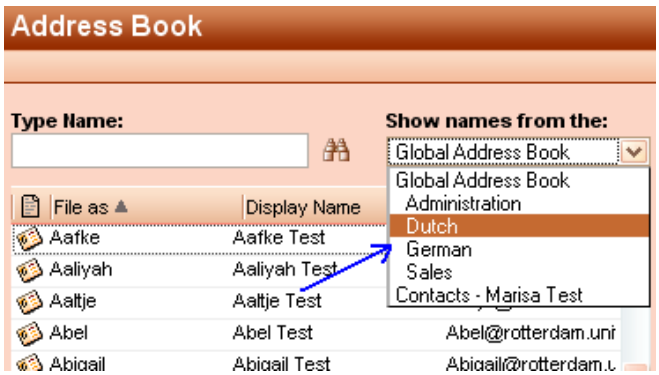


Figure 8.3. Listes d'adresses dans le carnet d'adresses

Pour définir une liste d'adresses dans Active Directory il est nécessaire d'utiliser le plugin ADS de Zarafa.

1. Sélectionner un dossier dans l'arborescence Active Directory à partir de la console des utilisateurs et groupes
2. Créer la nouvelle liste d'adresse avec **`Action > Nouveau > Liste d'adresses Zarafa`**
3. Insérer le nom de la liste d'adresses

4. Afficher les propriétés de la liste d'adresses nouvellement créée

5. Ajouter un filtre de recherche pour l'adresse, consulter *Section 8.6, « Exemples de critères LDAP »* pour des exemples de critères de recherche.

8.5.2.7. Cacher l'information du carnet d'adresse global avec ADS

Depuis ZCP 6.40 il est possible de cacher les utilisateurs, les contacts et les groupe du carnet d'adresses global. Pour empêcher les informations de s'afficher dans le carnet d'adresse global il faut cocher la case **`Cacher du carnet d'adresses`** sous l'onglet Zarafa dans Active Directory .

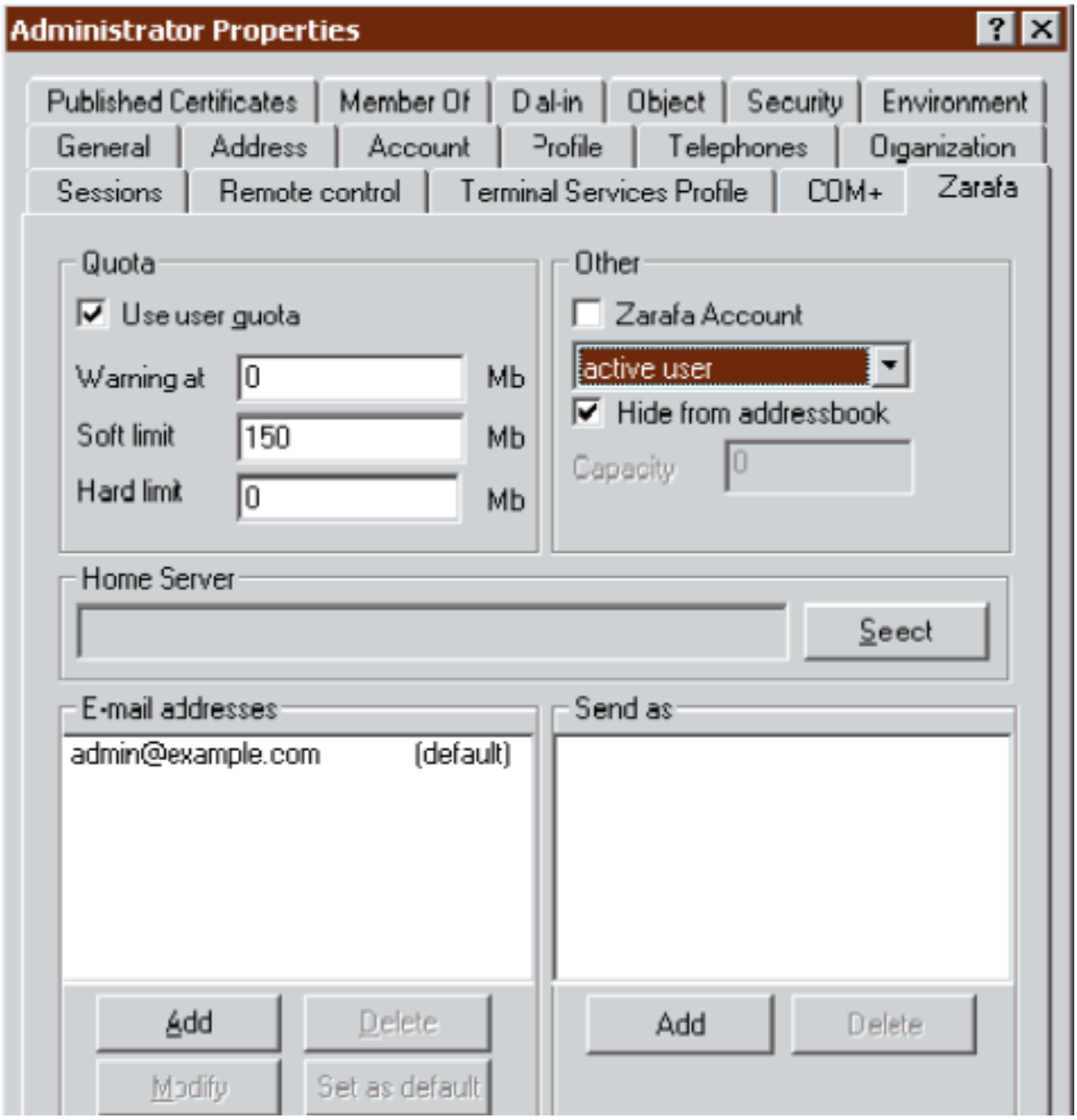


Figure 8.4. Cacher un utilisateur dans le carnet d'adresse global avec ADS

Note

L'utilisateur interne 'system' et le groupe 'everyone' peuvent être rendus invisibles à l'aide du fichier de configuration **`/etc/zarafa/server.cfg`**.

8.5.3. Gestion des utilisateurs avec OpenLDAP

8.5.3.1. Création des utilisateurs avec OpenLDAP

Les utilisateurs et les groupes peuvent être créés à l'aide des utilitaires d'administration OpenLDAP classiques, par exemple **`phpldapadmin`** ou bien l'utilitaire Windows **`ldapadmin`**.

Pour configurer les informations d'utilisateurs spécifiques à Zarafa, la classe d'objet 'zarafa-user' doit être ajouté à l'utilisateur. L'ajout de cette classe d'objet, permet d'ajouter des attributs Zarafa à l'utilisateur, tels que la définition des quotas, les permissions 'Envoyer en tant que', le type de boîte aux lettres.

8.5.3.2. Création des groupes avec OpenLDAP

Les groupes créés dans OpenLDAP seront utilisés par défaut comme groupes de sécurité par ZCP Les groupes de sécurité peuvent être utilisés afin de définir les permissions et pour l'envoi de courriel. Les groupes de diffusion ne peuvent être utilisés que pour envoyer du courrier et ne seront **`pas`** affichés lors de la configuration d'autorisations sur un dossier.

Pour basculer un groupe en simple groupe de diffusion, l'attribut **`zarafaSecurityGroup`** doit être défini sur **`0`**.

8.5.3.3. Création de contacts avec OpenLDAP

Le carnet d'adresses global peut recevoir des contacts. Les contacts sont généralement des adresses SMTP externes et ils peuvent être utilisés comme membres d'une lise de diffusion.

Les contacts doivent posséder le même attribut unique que les utilisateurs. Veuillez vérifier que **`ldap_unique_user_attribute`** soit correctement défini dans le fichier de configuration lap.cfg.

8.5.3.4. Configuration des permissions 'Envoyer en tant que' avec OpenLDAP

Les permissions 'Envoyer en tant que' peuvent être définies autant sur les utilisateurs que sur les contacts. Les utilisateurs et les groupes qui aspirent à pouvoir envoyer en tant qu'une adresse spécifique doivent être ajoutés à la liste des privilèges 'Envoyer en tant que'.

Pour vérifier si les permissions ont été correctement configurées, saisir la commande suivante :

```
zarafa-admin --list-sendas <username>
```

Par exemple :

```
zarafa-admin --list-sendas helpdesk
Send-as list (1) for user helpdesk:
       Username        Fullname
       ----------------------------
       john            John Doe
```

Les utilisateurs possédant les permissions 'Envoyer en tant que', devrait alors pouvoir ajouter l'autre adresse dans les champs‘De’ et ‘Envoyer en tant que’ depuis leur compte.

Depuis ZCP 6.40 le système 'Envoyer en tant que' a évolué :

- La configuration des permissions ‘Envoyer en tant que’ s'effectue d'une façon opposée à celle des versions précédentes de Zarafa. Les permissions ‘Envoyer en tant que’ doivent dorénavant être configurées sur l'utilisateur qui est sélectionné 'En tant que' dans le champ 'De'.
- Consulter *Section 3.5.1, « Étapes de mise à jour pré 6.40 »* pour la conversion des permissions 'Envoyer en tant que'.
- Les groupes peuvent maintenant également être utilisés pour configurer les permissions 'Envoyer en tant que'.

Note

Si des groupes sont employés pour les permissions 'Envoyer en tant que', il faut s'assurer que **`ldap_sendas_attribute_type`** soit défini sur **`dn`**. Ainsi que le montre la configuration LDAP suivante :

```
ldap_sendas_attribute = zarafaSendAsPrivilege
ldap_sendas_attribute_type = dn
ldap_sendas_relation_attribute =
```

8.5.3.5. Paramétrer les listes d'adresses dans OpenLDAP

Les listes d'adresses sont des sous-ensembles du carnet d'adresses global qui répondent à des critères spécifiques. Par exemple, vous pouvez créer une liste d'adresses qui contient tous les utilisateurs de Manchester et une autre contient tous les utilisateurs de Stuttgart.

Pour configurer une liste d'adresses dans OpenLDAP, il suffit de suivre ces étapes :

1. Créer une **`Unité organisationnelle`** pour toutes les listes d'adresses dans l'arborescence LDAP.
2. Créer un nouvel objet LDAP et ajouter la classe d'objet **`zarafa-addresslist`**
3. Définir l'attribut 'cn' sur le nom unique de la liste d'adresses
4. Créer un critère de recherche dans l'attribut **`zarafaFilter`**, consulter *Section 8.6, « Exemples de critères LDAP »* pour des exemples de critères de recherche.

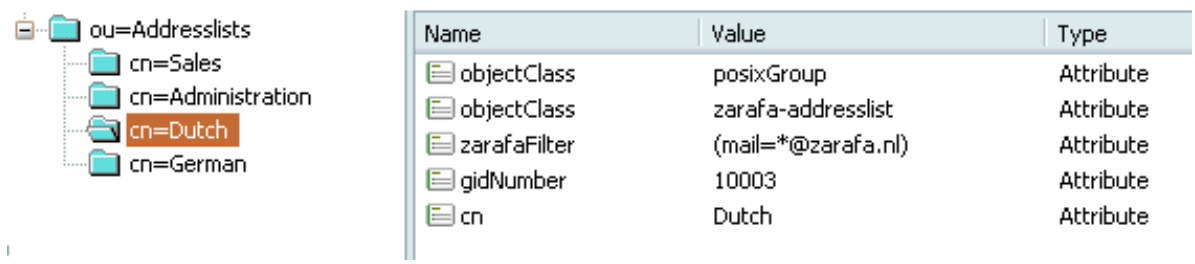


Figure 8.5. Liste d'adresses dans LDAP

Après le redémarrage de **`zarafa-server`**, les listes d'adresses devraient s'afficher dans le carnet d'adresses global.

8.5.3.6. Cacher l'information du carnet d'adresse global avec OpenLDAP

Depuis ZCP 6.40 il est possible de cacher les utilisateurs, les contacts et les groupe du carnet d'adresses global.

Pour empêcher les informations de s'afficher dans le carnet d'adresse global il faut définir l'attribut **`zarafaHidden`** dans OpenLDAP sur **`1`** pour l'objet spécifique que l'on souhaite cacher.

Note

L'utilisateur interne 'system' et le groupe 'everyone' peuvent être rendus invisibles à l'aide du fichier de configuration **`/etc/zarafa/server.cfg`**.

8.6. Exemples de critères LDAP

Un filtre LDAP doit être spécifié autant pour les listes d'adresses que pour les groupes dynamiques. Par exemple, le carnet d'adresses global comporte des utilisateurs hollandais et des utilisateurs allemands. Il est possible d'afficher ces utilisateurs en fonction de leur pays, en créant deux listes d'adresse dans l'arborescence LDAP. Tous les utilisateurs allemands ont le domaine*example.de* dans leur adresse électronique tandis que tous les utilisateurs hollandais ont *example.nl*.

Dans ce cas, le critère (**`mail=*@example.de`**) est utilisé pour la liste d'adresse allemande et le critère (**`mail=*@example.nl`**) pour la liste d'adresse hollandaise.

Toute combinaison avec des attributs LDAP est applicable. L'exemple suivant sélectionne touts les utilisateurs qui sont administrateurs Zarafa et qui possèdent la lettre**p** dans la valeur de **`cn`**.

```
(&(cn=*p*)(zarafaAdmin=1))
```

Ce prochain exemple sélectionne tous les utilisateurs utilisant l'adresse électronique *piet@example.com* ou *klaas@example.com*.

```
(|(mail=piet@example.com)(mail=klaas@example.com))
```

8.7. Gestion des fonctionnalités Zarafa

Certaines fonctionnalités offertes par ZCP peuvent être désactivées. Par défaut, toutes les fonctionnalités sont désactivées. Leur activation peut s'effectuer globalement ou au niveau de chaque utilisateur. Si une fonctionnalité a été globalement désactivée, elle peut cependant être activée individuellement au niveau de chaque utilisateur. Pour le moment, seules les fonctionnalités ‘imap’ et ‘pop3’ peuvent être contrôlées de la sorte.

Si la fonctionnalité ‘pop3’ est désactivée, les utilisateurs ne pourront pas se connecter à l'aide du protocole POP3. Il en est de même pour la fonctionnalité ‘imap’, ce qui de surcroit entraine une autre conséquence. Lorsqu'un utilisateur reçoit un courriel alors que la fonctionnalité ‘imap’ est activée, le courriel d'origine ainsi que plusieurs autres données d'optimisation IMAP seront stockés dans la base de données Zarafa et dans le répertoire contenant les pièces jointes. Ceci permet aux services IMAP fournit par zarafa-gateway d'être plus robustes. Par contre, ceci entraine également une plus grande utilisation de l'espace disque. La désactivation de la fonctionnalité ‘imap’ économisera ainsi l'espace disque.

Le tableau suivant indique lorsqu'un utilisateur peut utiliser l'IMAP ou le POP3.

Tableau 8.1. Affichage des contrôles d'accès

	Service activé pour l'utilisateur	Service désactivé pour l'utilisateur	Aucune configuration pour l'utilisateur
Service inclus dans disable_feature de server.cfg			
Service non-inclus dans disable_feature de server.cfg			

8.7.1. Activation globale des fonctionnalités

Pour activer une fonctionnalité spécifique, il suffit de modifier le paramètre **`disabled_features`** dans la configuration du serveur :

```
disabled_features = imap pop3
```

8.7.2. Dés/Activation des fonctionnalités au niveau de l'utilisateur

La gestion des fonctionnalités au niveau de l'utilisateur dépend du plugin utilisateur utilisé. Pour les plugins **`db`** et **`unix`**, l'utilitaire **`zarafa-admin`**sera utilisé :

```
zarafa-admin -u john --enable-feature imap
zarafa-admin -u john --disable-feature pop3
```

Pour les systèmes Active Directory ou OpenLDAP (c'est-à-dire avec l'utilisation des plugins **`ldap`** ou **`ldapms`**), les fonctionnalités seront gérées à l'aide des deux attributs LDAP **`zarafaEnabledFeatures`** et **`zarafaDisabledFeatures`**. Il faut s'assurer que les derniers fichiers de schema, ou le dernier plugin Active Directory, soit installés avant d'utiliser ces attributs. Ces attributs multivalués peuvent êtres constitués par toute chaîne de caractères, cependant, seules les fonctionnalités connues par Zarafa seront fournies au système.

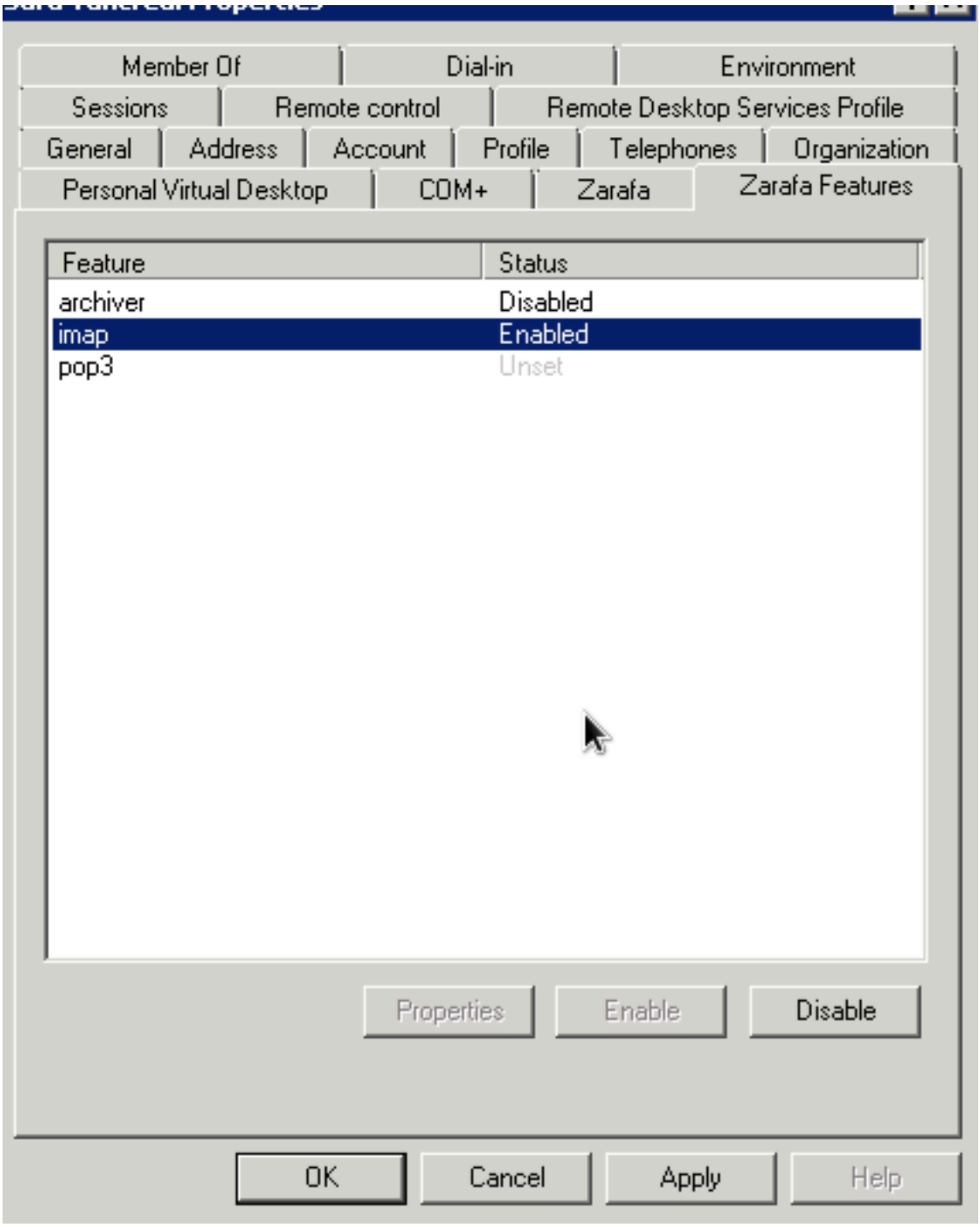


Figure 8.6. Onglet des fonctionnalités Zarafa dans ADS

Note

Il faut s'assurer qu'une fonctionnalité particulière ne soit pas enregistrée à la fois dans both zarafaEnabledFeatures et dans zarafaDisabledFeatures. Auquel cas il serait impossible de garantir une quelconque cohérence.

8.8. Configuration des ressources

ZCP gère la réservation automatique des ressources comme les vidéoprojecteurs, les salles de réunion ou tout autre équipement. Pour créer une ressource, il faut ajouter une nouvelle boîte aux lettres non-active ou bien sélectionner le type d'utilisateur ressource dans Active Directory ou dans OpenLDAP.

Avant qu'une ressource ne puisse être automatiquement réservé par des utilisateurs, elle doit être configurée pour accepter automatiquement les demandes de réunion. L'acceptation automatique des réunions peut se configurer de deux façons ; en employant l'utilitaire zarafa-admin ou bien en employant le client Outlook.

Pour configurer une ressource à partir d'Outlook, il suffit de suivre les étapes suivantes :

- Rendre la ressource temporairement active
- Se connecter en tant que la ressource sous Outlook
- Dans le menu Outils, cliquer sur Options, puis sur Options du calendrier.
- Sous Options avancées, cliquer sur Planification des ressources.
- Activer Accepter automatiquement les demandes de réunion
- Si la ressource doit décliner les doubles réservations, ou les réservations récurrentes, les options "Refuser automatiquement les demandes de réunion périodiques" et "Refuser automatiquement les demandes de réunion en conflit" devront être activées.
- Configurer les permissions sur l'agenda de la ressource, afin que les utilisateurs puisse la réserver. Les utilisateurs doivent posséder au minimum l'accès en écriture sur l'agenda de la ressource.

Pour configurer la ressource à l'aide de l'utilitaire zarafa-admin tool, veuillez saisir la commande suivante :

```
zarafa-admin -u <resource name> --mr-accept 1
```

La ressource acceptera alors automatiquement les réservations. Pour décliner les doubles réservations ou les réservations récurrentes :

```
zarafa-admin -u <resource name> --mr-decline-conflict 1
zarafa-admin -u <resource name> --mr-decline-recurring 1
```

Une fois que l'acceptation automatique des demandes de réunion a été configurée, il faut s'assurer que les utilisateurs possèdent au minimum l'accès en écriture sur l'agenda de la ressource. Les permissions peuvent être définies par un administrateur en ouvrant la boîte aux lettres de la ressource et en configurant les propriétés de son dossier Calendar.

Pour réserver automatiquement une ressource, il faut auparavant s'assurer que la ressource soit effectivement sélectionnée dans les plages de temps Libre/Occupé lors de la planification de la réunion.

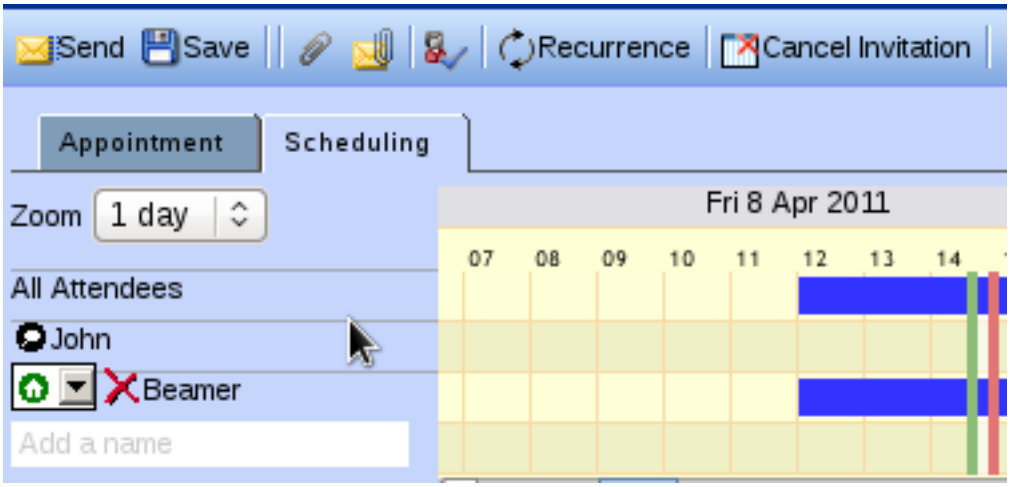


Figure 8.7. Affichage de la ressource dans les plages de temps Libre/Occupé

8.8.1. Méthodes de réservation des ressources

Il y a deux méthodes possibles pour réserver des ressources :

1. Réservation directe
2. Réservation par demande de réunion

Ces méthodes permettent toutes deux de réserver des ressources avec la même finalité : la ressource sera affichée comme occupée dans son agenda sur toute la plage de temps planifiée par l'utilisateur qui l'a réservée. Les deux méthodes gèrent toutes deux les refus pour conflit ou périodicité, mais la façon dont elles fonctionnent diffèrent sensiblement :

Tableau 8.2. Tableau de comparaison des méthodes de réservation de ressource

Réservation directe	Réservation par demande de réunion
Réservation directe dans l'agenda de la ressource	Envoi d'une demande de réunion qui entrainera une réponse
Nécessite un accès en lecture/écriture sur l'agenda de la ressource	Ne nécessite pas d'accès en lecture/écriture sur l'agenda de la ressource
Possibilité de limiter les utilisateurs pouvant réserver à l'aide des permissions	Impossible de limiter les utilisateurs pouvant réserver
Ne gère pas de multiple ressources utilisant le même agenda	Possibilité de définir la limite de double-réservation à 2 fois ou plus pour chaque ressource
Les utilisateurs externes ne peuvent pas effectuer de réservation	Les utilisateurs externes peuvent effectuer des réservations

8.8.1.1. Réservation directe

La réservation directe les la méthode par défaut employée par :

- Outlook 2000 - Outlook 2007
- Zarafa WebAccess

Pour qu'elle fonctionne, l'application cliente :

1. Ouvre l'agenda de la ressource

2. Vérifie les disponibilités restantes sur l'agenda
3. Crée un rendez-vous dans l'agenda
4. Notifie l'utilisateur que la ressource a été réservée

L'inconvénient principal de cette méthode, c'est que le client doit posséder un accès en écriture sur l'agenda de la ressource. Cela signifie par conséquent que l'utilisateur effectuant la réservation pourrait en théorie également planifier d'autres réservations dans l'agenda de la ressource sans adhérer à ses règles d'utilisation (p. ex. double réservation d'une salle).

Dans Outlook 2010, la méthode de réservation par défaut est passée à la réservation par demande de réunion. Celle dernière peut être réactivée au niveau de l'utilisateur en ajoutant la clé de registre suivante :

```
HKEY_CURRENT_USER\Software\Microsoft\Office\14.0\Outlook\Options\Calendar\EnableDirectBooking
 = (DWORD) 0x00000001
```

D'autres versions d'Outlook prennent également en charge la clé de registre permettant de *désactiver* la réservation directe.

Pour plus d'information à ce propos, veuillez consulter *http://support.microsoft.com/kb/982774*

8.8.2. Réservation par demande de réunion

Note

La réservation par demande de réunion a été ajoutée à partir de Zarafa 7.0.3. Les tentatives d'utilisation de cette méthode dans les versions précédentes entraineront l'impossibilité de confirmer les demandes de réservation des ressources et plus aucune réservation ne pourra s'afficher dans l'agenda de la ressource.

La réservation par demande de réunion fonctionne exactement comme l'envoi d'une demande de rendez-vous à un autre utilisateur. Lors de la réservation de la ressource, un utilisateur envoie une demande de réunion à la ressource par courriel. La ressource reçoit alors la demande, vérifie ses propres disponibilités et y répond exactement comme le ferait un utilisateur humain. L'utilisateur effectuant la réservation reçoit par courriel la réponse que la réunion a été *Acceptée* ou *Déclinée*.

Cela signifie que lorsque la réunion est envoyée aux participants, la ressource n'a pas vraiment déjà été réservée ; il est encore possible qu'un autre utilisateur finalise entre temps la réservation de cette même ressource, ce qui résulte alors en une demande *déclinée* dans la réponse de la ressource. L'organisateur doit alors à nouveau planifier la réservation et envoyer son actualisation à tous les participants.

L'avantage principal de cette méthode est que l'organisateur n'a pas besoin de posséder d'accès en lecture/écriture sur l'agenda de la ressource. Par ailleurs, cette méthode permet une gestion plus flexible des réservations. Par exemple, si il y a 5 vidéoprojecteurs qui doivent être créées en *resource*, alors, ils pourraient être créés en tant que 5 ressources séparées, qui chacune pourrait être réservée directement. Cependant, cela obligerait l'utilisateur à rechercher un vidéoprojecteur libre et à réserver spécifiquement celui là.

Avec la réservation par demande de réunion, l'administrateur peut tout simplement définir la capacité de l'équipement sur un chiffre supérieur à 1, par exemple 5 dans ce cas. L'administrateur n'a besoin que d'une seule ressource dotée d'une capacité de 5 pour représenter tous les vidéoprojecteurs.

Lorsque la demande de réunion est gérée par la ressource, elle vérifiera si *tous* les vidéoprojecteurs sont réservés au même moment et ne déclinera que si l'ensemble des 5 vidéoprojecteurs étaient alors déjà réservés.

Veuillez noter qu'il *faut* employer le type *equipment* pour la ressource si la notion de capacité doit lui être ajoutée. La capacité des ressources *room* est ignorée (il est impossible d'effectuer une double réservation de salle).

La réservation par demande de réunion est gérée par le script zarafa-mr-accept qui est installé par défaut. Ce script est déclenché par zarafa-dagent, à la fois en mode direct et en mode LMTP, lorsque le paramétrage mr-accept du destinataire est défini sur TRUE ET que le message entrant est une demande ou une annulation de réunion. Si le script zarafa-mr-accept échoue, le processus de distribution continue normalement, déclenchant éventuellement de règles de distribution et de messages d'absence du bureau.

8.8.3. Configuration de la méthode de réservation des ressources

Dans Outlook, la méthode de réservation peut être définie à l'aide de ce paramétrage

```
HKEY_CURRENT_USER\Software\Microsoft\Office\<OUTLOOK VERSION>\Outlook\Options\Calendar
\EnableDirectBooking = (DWORD) 0x00000001
```

Ce paramètre activera ou désactivera la réservation directe. La désactivation de la réservation directe implique l'utilisation de la réservation par demande de réunion.

Pour Zarafa WebAccess, la configuration de la méthode de réservation s'effectue à l'aide du paramètre

```
define('ENABLE_DIRECT_BOOKING', true)
```

dans le fichier config.php

Ce paramètre activera ou désactivera la réservation directe, et reflétera le comportement d'Outlook. Si la réservation directe est désactivée, la réservation par demande de réunion sera alors utilisée.

8.9. Out of office management

Users can normally manage their out of office replies from the Outlook, webclients and certain mobile devices. Sometimes users forget to turn on their out of office reply or out of office replies should be enabled for shared mailboxes.

For these purposes ZCP 7.1 is shipping a commandline utility to manage out of office replies.

To use the utility, use the following command:

```
zarafa-set-oof <username> 1|0 "Out of office subject" <path to out of office text>
```

To enable an out of office reply for the user john use:

```
zarafa-set-oof john 1 "I'm on holiday till the 30th of June" /tmp/oof.txt
```

8.10. Assistant de migration de boîtes aux lettres

L'assistant de migration de boîte aux lettres permet de déplacer les boîtes aux lettres entre différents nœuds d'un système multi-serveur.. L'utilitaire **`zarafa-msr`** doit être employé pour celà.

L'utilitaire **zarafa-msr** se connectera au serveur de gestion des utilisateurs (LDAP/AD) qui est défini dans le fichier de configuration server.cfg. Cet annuaire lui fournira les coordonnées du serveur qui gère l'utilisateur. Il se connectera alors à ce serveur et migrera la base de stockage entière de l'utilisateur vers le nouveau serveur devant l'accueillir tel que cela aura été défini dans le fichier de configuration msr. Une fois la migration achevée, l'utilitaire **zarafa-msr** conservera les deux bases de stockage synchronisées entre elles

L'utilitaire **zarafa-msr** n'effectue pas simplement la migration des éléments et des dossiers, mais également des permissions, des règles et des paramètres.

Note

L'utilitaire **zarafa-msr** n'est disponible que dans les systèmes multi-serveur. La gestion multi-serveur est disponible avec l'édition Enterprise et Zarafa en mode hébergé.

8.10.1. Conditions préalables

- Python 2.5 ou supérieur
- Liaison Python MAPI
- Zarafa 6.40.5 ou supérieur

8.10.2. Invocation

La seul argument requis par**zarafa-msr** est le fichier de configuration où sont consignées toutes les opérations de migration.

```
zarafa-msr msr.cfg
```

Une fois le déplacement des boîtes aux lettres effectué, **zarafa-msr** affichera le message suivant :

```
"x migrations have completed successfully, maintaining sync." (x migrations ont été
 effectuées avec succès, et sont maintenues en synchronisation"
```

x dénotant le nombre de boîtes aux lettres déplacées. L'administrateur peut désormais interrompre l'exécution de **zarafa-msr** en appuyant sur les touches Ctrl-C.

zarafa-msr peut être interrompu sans problème à tout moment en appuyant sur les touches Ctrl-C. À sa prochaine exécution, il reprendra sa tâche là où il avait été interrompu.

Si il n'est pas interrompu à l'aide des touches Ctrl-C, zarafa-msr continuera toujours la synchronisation.

L'utilitaire **zarafa-msr** peut être exécuté indifféremment depuis le serveur de destination ou depuis le serveur d'origine. Ou encore, de manière moins efficace, depuis n'importe quel nœud de l'architecture,multi-serveur.

Note

Il est conseillé de désactiver les quotas de boîtes aux lettres sur le serveur cible au cours de la migration.

8.10.3. Mise a jour LDAP/ADS

Il y a deux situations dans lesquelles il est possible de mettre à jour sans risque les serveurs des utilisateurs dont les boîtes aux lettres ont été déplacées :

1. **`zarafa-msr`** est toujours en cours d'exécution. Dans ce cas, toutes les modifications effectuées sur la boîte aux lettres d'origine seront propagées sur la nouvelle boîte aux lettres.

2. Aucune modification ne peut être effectuée dans la boîte aux lettres d'origine.

8.10.4. Configuration

Un fichier de configuration ordinaire rassemble généralement à ceci :

```
[Connection]
serverpath: file:///var/run/zarafa
sslkey_file: ssl.cert
sslkey_pass: pass

[Servers]

[Mapping]
user1: https://server2:237/zarafa
user2: https://server1:237/zarafa

[Logging]
log_file: /var/log/zarafa/msr.log
```

Note

In the directory /usr/share/doc/zarafa-multiserver/example-config an example msr.cfg can be found.

8.10.4.1. Section [Connection]

La section **`Connection`** comporte les informations permettant de se connecter à un nœud spécifique dans une grappe de serveurs. Cette section est obligatoire

Tableau 8.3. Options de la section Connection

Option	Valeur par défaut	Description
serverpath	*file:///var/run/zarafa*	Chemin d'accès au serveur N'importe quel nœud dans la grappe de serveur.
sslkey_file	-	Chemin d'accès vers le fichier de clé SSL.
sslkey_pass	-	Mot de passe pour la clé SSL key spécifiée dans l'option **`sslkey_file`**.
bidirectionnel	oui	Si activé, les modifications dans les boîtes aux lettres cibles seront synchronisée en retour.
force_source	no	Si activé, l'utilitaire zarafa-msr n'effectuera pas de redirection

Option	Valeur par défaut	Description
		vers le serveur d'origine à partir des informations LDAP.
workers	4	Nombre de processus concourant actif de synchronisation

8.10.4.2. Section [Servers]

La section **`Servers`** est une option facultative qui contient une liste d'alias de serveur. Ces alias peuvent être utilisés dans la section **`Mapping`** lorsqu'un nombre important de boîtes aux lettres sont déplacées vers le même serveur.

La section **`Servers`** n'a pas d'options pré-définies. Il faut plutôt utiliser le format

```
server_alias: server_path
```

Autant d'éléments que nécessaire pourront être placés dans cette section.

8.10.4.3. Section [Mapping]

La section **`Mapping`** contient la liste des noms d'utilisateurs ainsi que le nœud de destination de leurs boîtes aux lettres. Le nœud de destination peut être le chemin d'accès complet vers un serveur ou un des alias spécifié dans la section **`Servers`**.

La section **`Mapping`** n'a pas d'options pré-définies. Il faut plutôt utiliser le format

```
username: destination_node
```

Autant d'éléments que nécessaire pourront être placés dans cette section.

Pour déplacer une base de stockage publique un nom spécial devra être utilisé à la place du nom d'utilisateur : .

1. Dans un environnement multi-tenant, le nom du tenant pour lequel la base de stockage publique est déplacée devra être utilisé.

2. Dans un environnement comportant un unique tenant, le nom spécial __**`public`**__ devra être utilisé.

8.10.4.4. Section [Logging]

The **`Logging`** section is optional and contains logging specific settings. Currently the only setting is the **`log_file`** setting, which allows an alternate log file to be selected. By default a file called **`zarafa-msr.log`** will be created in the working directory.

8.10.5. Étapes post-migratoires

L'utilitaire **`zarafa-msr`** migrera des boîtes aux lettres complètes, tous paramètres inclus, vers le nœud de destination. Cependant, l'utilitaire **`zarafa-msr`** ne permettra pas de migrer l'état de synchronisation de l'utilisateur L'état de synchronisation est utilisé par les utilisateurs de Z-Push, de Blackberry ainsi que des utilisateurs déconnectés de Outlook.

Cela signifie que tous les utilisateurs de Z-Push devront réinitialiser leur appareil une fois la migration achevée. Sur certains appareils mobiles, une resynchronisation complète pourra être effectuée,

cependant, sur les iPhones ou les iPads, le profil Activesync complet devra être supprimé et recréé. Les utilisateurs possédant un appareil Blackberry devront être retirés puis ajoutés à nouveau dans la console d'administration de Blackberry Enterprise Server.

Les utilisateurs avec un profil Outlook hors-connexion bénéficieront du déclenchement automatique d'une resynchronisation à la suite de la migration msr. La resynchronisation réinitialisera l'état de synchronisation sur nouveau serveur, de sorte que toutes les modifications seront synchronisées sur le client Outlook.

Comme l'utilitaire **`zarafa-msr`** ne supprimera pas les boîtes aux lettres d'origine une fois la migration achevée, l'administrateur devra s'en charger. Sur le serveur d'origine, les commandes suivantes peuvent être utilisées afin de purger les boîtes aux lettres déplacées :

```
zarafa-admin --unhook-store <nom de l'utilisateur>
zarafa-admin --list-orphans
```

Enfin, utiliser le GUID de la base de stockage afin de la retirer complètement :

```
zarafa-admin --remove-store <GUID de la base de stockage>
```

Réglage des performances

Lors de l'installation d'un serveur Linux avec Zarafa, il est impératif que MySQL soit correctement configuré afin d'obtenir les performances maximales de votre serveur ; en effet, la plupart des goulots d'étranglement se situent au niveau de la base de données elle-même, il est donc très important de s'assurer que vos requêtes SQL soient effectuées aussi rapidement que possible.

De plus, pour les installations de taille importante, il est fortement recommandé de régler aussi les paramètres du cache Zarafa ; généralement, ceux-ci sont réglés assez bas afin de s'assurer que Zarafa puisse fonctionner sur des serveurs assez modestes, mais excepté dans le cas des installations les plus légères, ces paramètres par défaut doivent être revus à la hausse. Toute installation comportant plus de 50 utilisateurs devrait incontestablement régler les paramètres du cache en vue d'une performance maximale.

Ce document présume que le rôle principal du serveur est l'exécution de Zarafa. Il faut évidemment prendre d'autres facteurs en compte - par exemple, un système anti-spam ou un serveur Web fonctionnant avec un site différent de Zarafa WebAccess.

Des informations supplémentaires à propos du réglage des performances peuvent être consultées sur *http://wiki.zarafa.com*.

9.1. Considérations matérielles

Différentes configurations matérielles sont aussi à prendre en compte lors du paramétrage d'un serveur pour Zarafa. Nous verrons les différentes configurations matérielles pouvant influer sur les performances.

9.1.1. Utilisation de la mémoire

Le réglage de l'utilisation de la RAM est l'une des meilleures façons d'accroître les performances du serveur ; la mémoire RAM étant généralement peu coûteuse, son utilisation massive sur le serveur peut augmenter ses performances de façon exponentielle.

Par contre, un réglage augmentant de façon trop importante l'utilisation de la mémoire RAM peut inciter le serveur à sortir une partie de sa mémoire de la Swap, qui devra ensuite y être réincorporée, ce qui entraînerait un ralentissement généralisé du serveur. Il est donc primordial de configurer l'utilisation de la RAM par divers composants à l'aide d'un réglage assez élevé pour que toute la mémoire RAM disponible soit pleinement utilisée, sans toutefois dépasser un seuil excessif.

Afin d'utiliser la mémoire RAM de manière optimale, Zarafa a été conçu de manière à n'utiliser qu'une quantité fixe de la mémoire RAM physique; l'utilisation de la mémoire augmente à chaque nouvelle connexion d'un utilisateur, mais de manière marginale – la plus grande partie de l'utilisation de la mémoire étant déterminée par les réglages du cache dans le fichier de configuration. Ceci permet de contrôler très facilement la quantité précise de mémoire qui sera utilisée en situation opérationnelle, et on peut ainsi s'assurer que la quantité de mémoire RAM réellement utilisée ne puisse jamais dépasser les limites définies.

En résumé, l'utilisation optimale de la mémoire RAM consiste en un réglage aussi élevé que possible sans que cela n'incite le système à recourir de façon importante à la mémoire SWAP.

Il est très difficile de donner une valeur fixe à ce que serait une répartition optimale de l'utilisation de la mémoire pour un serveur particulier, du fait que le mode d'accès des données varie fortement d'un serveur à l'autre. Nous décrirons ici quelques méthodes empiriques et tenterons de clarifier les schémas d'utilisation de la mémoire RAM.

9.1.2. Considérations matérielles

Dans les serveurs tournant sous Zarafa, le goulot principal d'étranglement des performances se situe au niveau du passage des données depuis le disque dur du serveur et le temps nécessaire pour les faire parvenir jusqu'au client. Autrement dit, les performances I/O sont en général plus cruciales que les performances du CPU. En partant sur cette base, les conseils suivants peuvent se révéler utiles afin de sélectionner la configuration matérielle adéquate du système :

9.1.3. Plus de RAM c'est plus de rapidité

Une augmentation de la mémoire RAM se traduit par une mise en cache améliorée et par conséquent une rapidité plus importante.

Zarafa à été conçu spécifiquement pour utiliser au mieux les vastes quantités de RAM disponibles sur les serveurs modernes. Par contre, il ne faut pas oublier que dans un serveur Linux standard, la quantité utilisable maximale de RAM pour un serveur 32-bit est de 3 Go, à moins que la PAE (physical address extension) ne soit prise en charge par le noyau, le CPU et la carte mère. Si il est nécessaire d'utiliser plus de 3 Go de mémoire, sans aucune limitation, il faut alors utiliser une machine possédant un processeur en 64-bit, une distribution Linux en version 64-bit et un package Zarafa en version 64-bit.

9.1.4. RAID 1/10 est plus rapide que RAID 5

En général, une batterie RAID1 ou RAID10 est plus rapide en accès base de données que du RAID5. Si c'est possible, il faut toujours mieux opter pour du RAID10.

9.1.5. Une vitesse de rotation rapide (tr/min) engendre de meilleurs performances d'accès base de données

Les disques durs SCSI ou SAS de bonne facture ont habituellement des vitesses de rotation élevées de 10K voir 15K tr/min. La vitesse de rotation des disques affecte le temps de recherche sur les disques. Bien que le format de la base de données Zarafa soit optimisé de manière à avoir les données disponibles en mode série, et que la plupart des accès en lecture du disque soient limités autant que possible, le temps de recherche est néanmoins le facteur principal de vitesse en ce qui concerne l'I/O. Plus la vitesse de rotation est élevée et plus la recherche sera rapide.

9.1.6. Matériel RAID

Les contrôleurs matériels RAID ont souvent une large quantité de mémoire cache RAM. Ceci peut largement accroître les performances et le débit du sous-système I/O. Cependant, si un contrôleur matériel RAID est utilisé, il faut s'assurer soit de ne pas utiliser le cache en écriture différée, soit d'avoir un UPS fonctionnel et une procédure d'arrêt du serveur, car le cache en écriture différée sera perdu en cas de coupure de courant. Ceci n'est pas simplement dommageable pour les données en cours d'enregistrement à ce moment, mais les données InnoDB du disques pourraient également être affectées.

9.2. Utilisation de la mémoire

Il y a essentiellement 4 composants principaux du serveur qui utilisent la mémoire du système :

- Le cache cellule de Zarafa (met les cellules individuelles d'un tableau en mémoire)
- La taille du tampon MySQL (met en mémoire les accès en écriture et en lecture du fichier ibdata)
- Le cache de requête MySQL (met les requêtes identiques en mémoire)

Dans un serveur uniquement réservé à Zarafa, il faut s'assurer de définir ces caches en sorte qu'ils utilisent environ 80% de la mémoire RAM du serveur. Les 20% restants devant rester disponibles pour les processus du système, pour d'autres processus (comme le MTA) et pour le serveur Web.

En règle générale, on s'accorde à utiliser la répartition de mémoire RAM suivante :

Caches Zarafa :

- **cache_cell_size :** environ 25% de la taille totale de la mémoire RAM
- **cache_object_size :** environ 100 ko par utilisateur
- **cache_indexedobject_size :** environ 512 ko par utilisateur

Paramètres MySQL :

- **innodb_buffer_pool_size :** environ 25% de la taille totale de la mémoire RAM
- **mysql_query_cache :** 32 Mo
- **innodb_log_file_size :** 25% de innodb_buffer_pool_size
- **innodb_log_buffer_size :** 32 Mo
- **innodb_file_per_table**
- **table_cache: 1000**

These settings need to be configured in the **/etc/my.cnf** or **/etc/mysql/my.cnf** file below the [mysqld] section.

Avec ces réglages, environ 50%-60% de la mémoire RAM seront réservés aux caches de MySQL et de Zarafa. L'utilisation réelle de la mémoire par MySQL et par Zarafa sera en fait légèrement plus que cela, pour atteindre un total d'environ 80% de la taille de la mémoire RAM.

Veuillez consulter la documentation MySQL à propos de la configuration de **innodb_log_file_size** et des paramètres annexes, en effet, ces réglages devront être poussés un peu au-dessus de la normale afin d'accroître les performances en écriture. Cela n'affectera pas les performances en lecture.

Ces 4 réglages seront maintenant étudiés plus en détail afin d'illustrer les besoins de chacun des paramètres de ces caches.

9.2.1. Le cache cellule de Zarafa (cache_cell_size)

Les données qui sont actuellement montrées à l'utilisateur dans un affichage en tableau, passent par le *cache cellule*. Cela signifie que tout affichage d'un tableau dans Outlook ne récupère les informations depuis la base de données que si elles ne sont pas déjà contenues dans le cache. La durée de vie du cache est identique à la durée de vie du serveur, par conséquent, ouvrir une boîte de réception deux fois de suite devrait résulter en **0** accès disque pour le deuxième accès. Il est judicieux de régler le cache cellule aussi haut que possible, généralement à peu près identique à la taille du tampon MySQL.

9.2.2. Le cache objet de Zarafa (cache_object_size)

Le cache objet de Zarafa est utilisé pour mettre la table de hiérarchie en mémoire. Chaque objet auquel on accède sera placé dans ce cache, accélérant le prochain accès à la même information,

qui ainsi ne nécessite plus de requête vers la base de données. Plus le nombre d'éléments contenus dans les dossiers des utilisateurs est important et plus la taille de ce cache deviendra importante. L'information occupant une place relativement réduite, il n'est pas nécessaire d'allouer une taille importante à ce cache. Environ 1 Mo pour 10 utilisateurs représente même une surévaluation.

9.2.3. Le cache d'objet indexé de Zarafa (`cache_indexedobject_size`)

Pour ouvrir un élément spécifique, le programme doit envoyer une clé unique, intitulée **`entryid`**, au serveur qui a effectué la requête pour cet élément. Ce cache est un double index de la clé MAPI vers une clé de la base de données et vice versa. La transposition des clés est d'une grande importance. Ce cache est constitué par dossier, les dossiers de grande taille prendront donc la place d'autres informations autrement importantes. Il faut habituellement compter environ 0.5 Mo par utilisateur.

9.2.4. MySQL `innodb_buffer_pool_size`

Le tampon MySQL est utilisé afin de mettre en mémoire les accès en écriture et en lecture vers le fichier ibdata. Dans une machine dédiée MySQL, le réglage se situerait au choix entre 50% et 80% de la taille de la mémoire RAM physique de la machine. Si MySQL est exécuté à partir de la même machine que Zarafa, le réglage recommandé est d'environ 25% de la taille de la mémoire RAM physique (de telle sorte que le cache cellule de Zarafa puisse également être réglé à la même valeur)

9.2.5. MySQL `innodb_log_file_size`

Le réglage **`innodb_log_file_size`** défini la taille du fichier de journalisation des transactions. Par défaut il y a deux fichiers de journalisation. La taille conseillée pour **`innodb_log_file_size`** est d'environ 25% de **`innodb_buffer_pool_size`**.

9.2.6. MySQL `innodb_log_buffer_size`

La taille du tampon **`innodb_log_buffer_size`** utilisé par InnoDB pour consigner les fichiers de journalisation sur le disque. Un tampon de journalisation important permet à des transactions de grande taille d'être effectuées sans avoir à consigner la journalisation sur la disque avant la fin des transactions. Si de larges transactions sont effectuées, agrandir la taille du tampon de journalisation minimisera l'I/O disque. Cette valeur devrait être environ 25% de **`innodb_log_file_size`**.

9.2.7. MySQL `query_cache_size`

Le cache des requêtes MySQL est habituellement désactivé. l'activation du cache des requêtes peut résulter en un gain modeste de performance, cependant, l'augmenter de plus de quelques Mo de mémoire est inutile puisque la récurrence des mêmes requêtes SQL est plutôt faible.

9.2.8. MySQL `innodb_file_per_table`

L'option innodb_file_per_table permet de créer un fichier de données innodb pour chaque table de la base de données, plutôt qu'un unique large fichier ibdata pour toutes les données. L'utilisation d'un fichier pour chaque table apporte plus de flexibilité en permettant le déplacement des tables vers des partitions ayant d'autres systèmes de fichiers afin d'améliorer les performances.

9.3. Configuration des modules sur différents serveurs

Certains composants du serveur Zarafa peuvent être hébergés sur des machines différentes. En fait, pratiquement toutes ses différents composants peuvent être exécutés sur des systèmes différents.

Cependant, en pratique, la répartition de tous les modules de Zarafa sur plusieurs serveurs séparés, n'améliore pas les performances. Les principaux composants qui méritent réflexion sont les suivants :

- *Server1 :* le serveur MySQL
- *Server2 :* le serveur Zarafa
- *Server3 :* le MTA + l'AntiSpam/AntiVirus
- *Server4 :* le serveur Web

Si chacun de ces 4 composants était hébergé sur 4 serveurs différents, chaque serveur communiquerait avec les autres comme s'il n'y avait qu'un système unique. Cette configuration peut être mise en œuvre assez simplement en paramétrant ces différents composants du système afin qu'ils puissent communiquer avec un autre serveur.

Le serveur MySQL n'a besoin d'être accessible que par le processus **`zarafa-server`** du *Server2*. Ceci peut se faire très simplement en définissant les paramètres corrects de l'identifiant et de l'ordinateur hôte dans le fichier **`server.cfg`** de Zarafa.

Le serveur Zarafa lui-même sera contacté par les clients Outlook, par le *Server3* (MTA), et par le *Server4* (serveur Web). Ceci se fait automatiquement puisque le processus **`zarafa-server`** est à l'écoute du port **`236`**sur le *Server2*, et que tous les autres serveurs peuvent s'y connecter.

Le *Server3*acceptera le courrier électronique sur le port **`25`** ou ira le récupérer à l'aide d'un protocole de courrier POP3 ou autre. Après avoir été scannés à l'aide d'un anti-spam et d'un anti-virus, les courriels seront transmis au processus **`zarafa-dagent`**. Le processus **`zarafa-dagent`**peut être configuré pour se connecter avec un certificat SSL au *Server2*. Ce certificat SSL est requis afin que **`zarafa-dagent`** puisse se connecter sur le port **`236`**. Lorsque cette configuration est réalisée à la fois sur le *Server3* et sur le *Server2*, le courrier pourra être livré directement au *Server2* par le *Server3*.

Le *Server4* est le serveur Web, il exécute Apache et il accepte les connexions du port **`80`** (ou **`443`** en SSL). WebAccess de Zarafa peut être configuré (dans **`config.php`**) afin de se connecter sur le port **`236`** (ou port **`237`** en SSL) du *Server2* pour accéder aux données. Une fois cette configuration effectuée, le serveur Zarafa est prêt à l'usage. Aucun paramétrage supplémentaire n'est nécessaire.

Sauvegarde & Restauration

Actuellement, Zarafa fournit trois solutions de restauration d'éléments :

- Through the softdelete restore system
- En utilisant le système de sauvegarde par boîte individuelle (brick-level)
- Avec la sauvegarde intégrale de la base de données

10.1. Softdelete restore

The softdelete restore can be used by users from Outlook with the *Restore deleted items* dialog from the *Tools* menu to restore deleted items. This will cover most accidental deletions.

Les éléments qui sont supprimés par un utilisateur (en vidant le répertoire 'Éléments supprimés' ou même par une suppression physique en utilisant les touches Maj-Suppr dans Outlook), sont simplement placés dans le cache de suppression logique. Cela signifie que ces éléments ne seront pas réellement retirés de la base de données avant que le délai de rétention ne soit expiré. Cette période limite d'expiration peut être spécifiée dans le fichier **`server.cfg`** et est définie à **`30`** jours par défaut.

Noter que l'option *Restaurer les éléments supprimés* ne s'applique que sur le répertoire actuellement sélectionné.

Le tableau suivant affiche les différentes solutions possibles selon le statut de l'initiateur de la sauvegarde, et les situations auxquelles elles sont le mieux adaptées.

Tableau 10.1. Options de récupération

Requête de restauration	% de temps effectué	Solution de sauvegarde	Initiateur
Éléments < 30 jours	80 %	Suppression logique	Utilisateur et Administrateur
Éléments >= 30 jours	10 %	Boîte individuelle	Administrateur
Éléments d'un utilisateur spécifique	5 %	Boîte individuelle	Administrateur
Éléments sur une période de temps donnée	3 %	Boîte individuelle	Administrateur
Restauration après un sinistre	2 %	Dump MySQL	Administrateur

Ainsi, les requêtes de restauration les plus communes peuvent être effectuées par l'utilisateur lui-même. C'est parce que le système de suppression logique est accessible directement à partir d'Outlook.

Si des messages plus anciens doivent être restaurés, l'administrateur devra consulter les sauvegardes effectuées. Il est impossible de restaurer un élément individuel à l'aide d'un dump MySQL, c'est dans ce cas que l'utilitaire **`zarafa-backup`** sera utile.

La sauvegarde par boîte individuelle de l'utilitaire **`zarafa-backup`** ne contient cependant pas toutes les informations nécessaires pour effectuer une restauration à la suite d'un sinistre. Un 'dump' complet de la base de données MySQL devra être exécuté pour pouvoir effectuer ce type de restauration.

10.2. 'Dump' complet de la base de données

Toutes les données enregistrées par le serveur Zarafa sont stockées dans une base de données MySQL. Cela signifie que pour effectuer une restauration en cas de sinistre, il suffit d'effectuer une restauration complète de la base de données en question. Ceci peut être accompli de plusieurs façons différentes, cependant nous n'aborderons ici que deux solutions efficaces. Attention, il existe également plusieurs façons de ne pas effectuer une sauvegarde.

10.2.1. Générer un 'dump' SQL à l'aide de mysqldump

Le contenu entier d'une base de données Zarafa peut être enregistré dans un fichier à l'aide de la commande **`mysqldump`**. Il y a cependant dans ce cas des paramétrages importants à respecter : il faut toujours spécifier l'option **`--single-transaction`** à **`mysqldump`**. Ainsi, **`mysqldump`** créera sur le disque un cliché unique de la base de données. Ceci garantira que tout nouvel accès en écriture effectué sur la base de données pendant la procédure de sauvegarde ne sera pas inclus. Concrètement, le dump effectuée est un 'cliché' de la base de données au moment précis où le dump est lancé.

Lors de l'utilisation de **`mysqldump`**, il est très important de ne pas effectuer de verrouillage de table. Ceci signifie que l'option **`--opt`** et l'option **`--lock-tables`** ne doivent jamais être utilisées au cours d'une sauvegarde 'dump' de la base de données Zarafa. En effet, ces options 'verrouillent' les tables pendant leur déchargement sur le disque, causant ainsi le blocage de tout accès à la base de données durant toute la procédure de sauvegarde. Premièrement ce n'est pas nécessaire et deuxièmement cela peut causer le rejet des courriers électroniques arrivant au cours la procédure (selon la configuration du MTA).

Un simple :

```
mysqldump --skip-opt ---single-transaction -p <database> > <dumpfile>
```

créera un 'dump' complet de la base de données.

10.2.2. 'Dump' binaire de données à l'aide de LVM Snapshotting

Cette technique utilise la fonctionnalité 'LVM Snapshot' afin de 'figer' efficacement un cliché binaire du fichier de la base de données, alors même que la base est en cours d'utilisation. Ce cliché 'figé' est alors simplement copiée en mode binaire sur un serveur distant. Cette solution fonctionne car InnoDB s'assure qu'un cliché unique de la base de données soit toujours cohérent (p.ex. Il sera toujours en mesure de restaurer la base de données lorsque MySQL est démarré avec cet ensemble de données.)

Du fait que l'installation et la configuration de LVM relèvent tous deux d'un processus complexe, nous invitons l'utilisateur à se référer à la documentation et aux utilitaires LVM permettant de monter un volume LVM dédié à une base de données MySQL, et permettant de créer et de supprimer des partitions images.

10.2.3. Sauvegarde des pièces jointes

Lorsque les pièces jointes sont stockées en dehors de la base de données, il faut s'assurer que ces pièces jointes soient également mises en sauvegarde.

Plusieurs méthodes de sauvegarde peuvent être utilisées afin d'effectuer des copies de secours pour les pièces jointes :

- Rsync

- Copier tous les fichiers vers un serveur de secours externe ou vers une disque dur externe
- Avoir recours à une solution de sauvegarde (commerciale) Linux, telle que SEP, Bacula, Arkeia ou autre

10.3. Sauvegarde par boîte individuelle

Les éditions commerciales de ZCP fournissent un utilitaire de sauvegarde par boîte individuelle (brick-level). Cet utilitaire créera une sauvegarde de chaque boîte aux lettres sur des fichiers séparés. La seconde fois qu'une sauvegarde sera effectuée, seuls les nouveaux éléments ainsi que les éléments modifiés seront ajoutés à la sauvegarde.

Veuillez noter que ce type de sauvegarde n'est pas conçu pour une restauration suite à un sinistre. En effet, seuls les éléments sont enregistrés dans la sauvegarde. Aucunes informations à propos de l'utilisateur, ni celles qui ont été créées par les utilisateurs, telles que des règles, n'y sont enregistrées.

10.3.1. Format de sauvegarde

L'utilitaire de sauvegarde génère 2 fichiers pour chacune des boîtes aux lettres : un fichier de données et un fichier d'indexation.

Le fichier d'indexation contient les informations de dossiers, de hiérarchie et de messages. Les champs sont séparés par deux-points de ponctuation. Il y a 3 types d'entrées dans le fichier d'indexation, qui sont **`R`**, **`C`** et **`M`**. L'entrée **`R`** signifie **`Root`**, et est toujours la première et la seule entrée **`R`** de l'index. Elle contient une clé utilisée par les dossiers comme clé parent pour dénoter qu'ils sont directement connectés au conteneur Root de la base de stockage.

L'entrée **`C`** signifie **`Conteneur`**, qui peut être tout type de fichier. Elle possède 2 clés, une clé parent et une autre clé permettant d'identifier le conteneur lui-même. Elle possède également une clé de restauration unique. Cette clé peut être utilisée afin de sélectionner le dossier pour l'utilitaire de sauvegarde. Il y a un compteur du nombre d'éléments contenus dans le dossier, un horodatage Unix marquant la dernière modification effectuée, et le type de dossier (p. ex. **`IPF.Note`** pour un répertoire de courrier, **`IPF.Appointment`** pour un agenda). La dernière partie d'une entrée **`C`** est composée du nom du répertoire, qui peut lui-même contenir deux-points de ponctuation, dénotant ainsi que c'est la dernière partie de l'entrée. Une liste détaillée des champs d'un **`Conteneur`** est disponible dans l'annexe.

L'entrée **`M`** de l'index signifie **`Message`**, qui peut être tout type de message ou d'élément. Elle a une clé parente, qui correspond à la clé d'un dossier. Elle a également une clé de restauration qui peut être utilisée afin de restaurer ce message spécifique. Il y a ensuite un horodatage Unix qui marque la dernière modification effectuée sur le **`message`**. Si un utilisateur modifie de nouveau ce message, l'horodatage sera mis à jour. L'entrée de l'index continue avec le type de **`message`** (courrier électronique, agenda, demande de réunion, etc). L'entrée comporte un emplacement flottant, là où l'élément commence dans le fichier de données, et en dernier, elle comporte le sujet de l'élément. Comme ce sujet peut contenir deux-points de ponctuation, il est positionné à la fin de l'entrée. Une liste détaillée des champs d'un **`Message`** est disponible dans l'annexe.

Le fichier des données est un cliché binaire de toutes les propriétés des messages, de leurs destinataires et de leur pièces jointes. Les dossiers sont définis uniquement dans le fichier d'indexation et par conséquent, seul leur nom est sauvegardé, puisque cela est suffisant pour pouvoir les recréer.

10.3.2. Procédure de sauvegarde

Au cours de la première sauvegarde d'une base de stockage, l'utilitaire de sauvegarde exécutera les actions suivantes :

- Création d'une liste de tous les dossiers d'une base et de leur contenu
- Tout élément trouvé sera enregistré sur le disque

Du fait qu'une liste de tout ce que se trouve dans la base de stockage doit être initialement générée, les nouveaux éléments créés au cours de cette procédure seront ignorés et ne seront pas sauvegardés. Les élément déplacés seront présents dans la sauvegarde, mais à leur emplacement initial. Les éléments supprimés de manière définitive au cours de la sauvegarde ne seront pas enregistrés puisqu'il sera impossible de les ouvrir.

Lorsqu'une nouvelle sauvegarde sera à nouveau lancée, la sauvegarde précédente sera utilisée et une mise à jour incrémentielle sera automatiquement effectuée, et elle mettra en œuvre les actions suivantes :

- Lecture du fichier d'indexation et création de l'arborescence de la sauvegarde précédente
- Création d'une liste de tous les dossiers d'une base et de leur contenu
- Pour chaque conteneur, localisation des éléments qui ont déjà été sauvegardés et qui n'ont pas été modifiés, suivi par leur retrait de la liste.
- Suppression de l'ancien fichier d'indexation
- Sauvegarde des éléments toujours présents dans la liste, suivi de leur ajout dans le fichier des données

Pour démarrer le processus de sauvegarde, il suffit d'exécuter :

```
zarafa-backup -u <nom_de_l'utilisateur>
```

ou pour tous les utilisateurs et les dossiers publiques :

```
zarafa-backup -a
```

To speed up the backup process multiple threads can be configured in the **`backup.cfg`**. The default option is 1 thread, so for larger environment increasing this number is recommended.

La méthodologie employée par l'utilitaire de sauvegarde entraîne plusieurs observations. Lorsque la liste de l'index précédent est comparée avec le contenu actuel de la base, cette comparaison est effectuée par conteneurs correspondants. Ainsi, si un utilisateur déplace des éléments d'un dossier à l'autre, ils ne seront pas trouvés dans l'index et seront donc sauvegardés à nouveau car ils seront marqués comme étant 'nouveaux' dans le dossier où ils ont été déplacés.

Si un message a été modifié par un utilisateur depuis la dernière sauvegarde, l'élément aura une nouvelle 'date de dernière modification', et sera de nouveau sauvegardé dans sa totalité car autrement, le processus de sauvegarde deviendrait péniblement lent s'il devait vérifier chacune des propriétés du message afin de différencier celles qui ont été modifiées de celles qui ne l'ont pas été. Écraser l'ancien message peut également se révéler problématique puisque le nouveau message peut être plus volumineux que l'ancien et peut dans ce cas ne pas loger dans la place laissée par l'ancien message.

Then when the actual backup process starts, it will first remove the old index. The index file will then be rebuild while the backup processes each message found in the list. The changed data will be placed in a new data file with an incrementing counter in its filename, keeping the old information which was still available and did not need to be stored again.

Pour consulter toutes les options de l'utilitaire **`zarafa-backup`** exécuter :

```
man zarafa-backup
```

10.3.3. Procédure de restauration

Pour restaurer des éléments sauvegardés avec l'utilitaire **zarafa-backup**, il suffit d'utiliser l'utilitaire **zarafa-restore**. Pour restaurer des éléments ou des dossiers entiers, vous devez auparavant déterminer la clé de restauration correspondante dans le fichier d'indexation **user.index.zbk**.

This index file isn't humanly readable with a text editor. Instead, use the **readable-index.pl** Perl script, which can be found in **/usr/share/zarafa-backup/**. To identify items, use the folder name field or the subject to find the items needed to be restored.

```
/usr/share/zarafa-backup/readable-index.pl username.index.zbk
```

When the items are found, place the restore keys in a separated file, or give them as parameters to the **zarafa-restore** tool. If the restore key of a folder is entered, the complete folder with all its items will be restored on one level. If the sub folders of the selected folder need to be restored, add the **-r** parameter to the command. The following example restores the inbox with sub folders from **userA**. The restore key **AF000000** is found in the **userA.index.zbk** file and needs to be defined at the end of the command.

```
zarafa-restore -u userA -r -c userA.index.zbk AF000000
```

L'utilisation du paramètre**--c** comme référence au fichier d'indexation n'est pas nécessaire lors de l'utilisation d'un fichier d'indexation provenant du même utilisateur. Par exemple, dans le cas suivant : **zarafa-restore --u userA**, l'utilitaire **zarafa-restore** utilisera automatiquement le fichier d'indexation **userA.index.zbk** si le fichier **index.zbk** est présent dans le répertoire à partir duquel la commande est exécutée.

Dans l'exemple suivant, un fichier (**keys.txt**) contenant plusieurs clés de restauration pour de multiple éléments et dossiers appartenant à l'utilisateur **userA** est utilisé. Chaque clé de restauration du fichier doit être séparé des autres par une nouvelle ligne.

```
zarafa-restore -u userA --r --i keys.txt
```

Pour effectuer la restauration complète d'une boîte aux lettres d'un utilisateur, le script suivant peut être utilisé.

```
/usr/share/zarafa-backup/full-restore.sh <username>
```

Veuillez vous assurer que le script est exécuté à partir du dossier de restauration. Pour effectuer la restauration complète d'une boîte aux lettres d'un autre utilisateur, exécuter :

```
/usr/share/zarafa-backup/full-restore.sh <username> <destination_username>
```

Pour toutes les options de l'utilitaire **zarafa-restore**, veuillez consulter la page man.

```
man zarafa-restore
```

BlackBerry Enterprise Server

11.1. Conditions préalables

ZCP fonctionne avec BlackBerry Enterprise Server 4 ainsi qu'avec BlackBerry Enterprise Server 5 (Express), cependant il est recommandé d'utiliser Blackberry Enterprise Server 5 qui est plus récent.

11.1.1. Logiciels

Pour pouvoir utiliser BlackBerry Enterprise Server (BES) avec Zarafa, les logiciels suivants doivent être installés :

- Client Zarafa version 6.40.5 ou supérieure
- Connecteur Zarafa BES
- BlackBerry Enterprise Server 5 ou Blackberry Enterprise Server Express pour MS Exchange
- Microsoft Outlook 2003 ou 2007
- Microsoft CDO (dans le pack d'installation pour Office 2003 ou en téléchargement séparé pour Office 2007)

Un serveur ZCP version 6.30.18 ou 6.40.0 ou supérieure, correctement configuré et en fonctionnement est également indispensable.

11.1.2. Préparation de l'authentification

Un certificat de confiance est nécessaire pour établir une communication entre le composant agenda de BES (**`CalHelper.exe`**) et Zarafa. Pour une communication ordinaire (courriel) la seule exigence est d'avoir un utilisateur possédant les privilèges administrateur. Un compte administrateur préexistant peut être utilisé dans ce but, mais il est également possible de créer un nouveau compte administrateur, généralement *besadmin*.

Pour créer un certificat SSL, suivre les étapes décrites dans *Chapitre 6, Configurations avancées*. Un certificat est nécessaire. Copier la clé privée (p. ex. **`bes.pem`**) vers la machine Window qui exécute BES, et placer la clé publique (p. ex. **`bes-public.pem`**) dans le répertoire **`/etc/zarafa/sslkeys`** du serveur Zarafa.

Important

Si un certificat auto-signé est utilisé (ce qui est très probable), alors, Outlook DOIT être démarré par le compte utilisateur employé par BES et se connecter au serveur une **seule** fois à l'aide du protocole SSL. Un pop-up d'avertissement SSL s'affichera avec l'option *se rappeler de ce choix*. Si cette option n'est pas cochée, il y aura ensuite des problèmes de synchronisation avec l'agenda. Si une grappe de serveurs est utilisée, il faudra se connecter à chaque serveur.

11.2. Étapes de l'installation

Note

Si un serveur BES4 existant est en cours de remplacement, veuillez vous assurer que l'ancien répertoire **`CalHelper.exe.local`** est **supprimé**, car il n'a plus aucune utilité pour cette version.

Important

BES 5.0 **nécessite** un serveur Active Directory pour son installation. Cependant, ce sera nécessaire uniquement pendant l'installation, ce ne sera plus nécessaire après la mise en route du serveur BES. Par ailleurs, la machine accueillant BES5 **doit** faire partie du domaine, même si tout peut être installé en utilisant le compte d'un *administrateur* local. Si un seul de ces éléments est manquant, l'installation ne pourra pas s'effectuer complètement.

1. S'assurer que le serveur ZCP est correctement configuré pour SSL (voit l'étape précédente).
2. Installer Outlook (Dans Outlook 2003, choisir le mode d'installation personnalisée afin de pouvoir activer CDO).
3. Installer CDO (seulement nécessaire pour Outlook 2007).
4. S'assurer de copier **`cdo.dll`** et **`gapi32.dll`** depuis **`c:\program files\common files\system\msmapi\langid`** vers **`c:\windows\system32`**, autrement le serveur Blackberry ne pourra pas détecter CDO.
5. Installer le client Windows de Zarafa.
6. Installer le connecteur BES de Zarafa.
7. **`Démarrer`** → +Zarafa+ → +Zarafa BES connector+ → +Créer profil MAPI+. L'adresse du serveur Zarafa, le nom d'utilisateur et le mot de passe seront demandés. Un compte administrateur devra alors être spécifié afin de créer le profil. Il est recommandé d'utiliser SSL tout de suite, afin de détecter tout problème de configuration SSL dès le départ.
8. Trouver tous les fichiers dans la machine se nommant **`ems*32.dll`** (généralement **`emsui32.dll`**, **`emsmdb32.dll`** et **`emsabp.32.dll`**) puis **remplacer** chacun d'entre eux par le fichier **`emsmdb32.dll`** qui est fourni dans répertoire 'program files' du connecteur BES de Zarafa.
9. Définir le chemin d'accès adéquat ainsi que le mot de passe pour la clé SSL et l'adresse du serveur dans le fichier de configuration **`C:\Program Files\Zarafa\Zarafa BES Connector\exchange-redirector.cfg`**
10. S'assurer que toutes les étapes (1-10) ont été suivies et redémarrer la machine.
11. Lancer l'installeur BES.
12. Choisir d'ignorer l'avertissement qui s'affichera à propos des bibliothèques MAPI requises.
13. Le compte administrateur requis au cours de l'installation BES doit être l'administrateur du serveur Active Directory et du domaine Windows.

14. Utiliser l'adresse du serveur Zarafa et le compte administrateur de l'étape 7 lorsque le serveur Exchange et le compte de boîte aux lettres vous seront demandés.

15. Choisir d'ignorer l'avertissement qui s'affichera à propos des privilèges *Administrateur Exchange Affichage seul*.

16. Pour se connecter à l'interface d'administration BES, utiliser l'authentification BES si le serveur Active Directory n'a été mis en service que temporairement afin d'installer BES

L'installation devrait alors s'achever normalement et les services Blackberry démarreront automatiquement.

Note

Il sera impossible de contacter tout serveur Exchange à partir de cette machine après avoir modifié les fichiers **`ems*32.dll`**.

11.3. Erreurs BES

La plupart de problèmes pouvant survenir sont les suivants :

- Une mauvaise configuration SSL sur le client (erreurs MAPI_E_INVALID_ARG dans le fichier de journalisation *MAGT* : certificat SSL ou mot de passe incorrect).
- Une mauvaise configuration SSL sur le serveur (erreurs MAPI_E_NETWORK_ERROR dans le fichier de journalisation *MAGT*).
- Le certificat SSL du serveur n'est pas accepté pour le compte utilisé (erreurs MAPI_E_NETWORK_ERROR dans le fichier de journalisation *MAGT*). Pour résoudre le problème, démarrer une fois Outlook en utilisant SSL et se connecter à tous les serveurs de la grappe.
- Le fichier de journalisation *MAGT* se plaint à propos du profil BlackBerryServer manquant. PR_PROFILE_USER ou PR_PROFILE_HOME_SERVER_DN : Le profil BlackBerryServer de BES doit être recréé à partir de l'élément du menu suivant : *Start+→+Zarafa+→+Zarafa exchange redirector+→+Create BES profile*.
- Le fichier de journalisation *MAST* se plaint de ne pouvoir mettre à jour la liste des utilisateurs GAB : La version 6.30.18 ou 6.40.0 ou supérieure doit être sur le serveur.

Note

D'autres techniques et méthodes supplémentaires d'intégration ZCP BlackBerry peuvent également être consultées sur *http://www.zarafa.com/wiki/index.php/Blackberry_integration*.

Annexe A; Stratégies de mise à jour pré 5.2x

12.1. Mise à jour de la base de données depuis la version 4.1 ou 4.2

Avant que Zarafa ne puisse être redémarré, la base de données doit être mise à jour. Plusieurs scripts sont requis, selon la version depuis laquelle la mise à jour est effectuée. Ces scripts de mise à jour ne sont nécessaires que dans le cas d'une mise à jour depuis une version 5.0x ou antérieure. Les scripts disponibles sont les suivants :

```
db-convert-4.1-to-4.2
```

Ce script Perl met à jour le format 4.1 de la base de données vers le format 4.20. Ces changements affectent la façon dont les utilisateurs sont stockés dans la base de données. L'exécution de ce script est indispensable et doit être effectuée de la manière suivante :

```
perl /usr/share/doc/zarafa/db-convert-4.1-to-4.2 \
       <dbuser> <dbpass> <dbname>
```

Remplacer **<dbuser>** par le nom de l'utilisateur avec lequel la connexion à la base de données est effectuée. Remplacer **<dbpass>** avec le mot de passe de l'utilisateur de la base de données. Si il n'y a pas de mot de passe, substituer 2 apostrophes **''** à la place. Remplacer **<dbname>** avec le nom de la base de données. Ce qui résultera en une commande ressemblant à celle-ci :

```
perl /usr/share/doc/zarafa/db-convert-4.1-to-4.2 root '' zarafa
```

```
db-convert-4.20-to-4.21
```

Ce script Perl met à jour le format 4.20 de la base de données vers le format 4.21. Il remplacera une clé d'indexation afin d'améliorer la rapidité de la base de données. Il est vivement recommandé d'utiliser ce script, qui doit être exécuté de la même façon que le script **db-convert-4.1-to-4.2** précédent.

En fonction de la taille de la base de données et de la vélocité du système, l'exécution de ce script peut prendre un certain temps, probablement de l'ordre de 10 à 30 minutes.

```
db-convert-4.20-to-innodb.sql
```

Ce script SQL convertit la base de données 4.20 vers le format InnoDB. Les installations, depuis la version 4.0, créaient des tables MyISAM. Cependant, la disposition de la base de données SQL actuelle est optimisée pour le format InnoDB. C'est pourquoi la conversion de la base de données MyISAM vers InnoDB apporte une accélération importante des performances. De plus, le format InnoDB est moins sujet aux erreurs, et utilise moins de verrouillage global de table. Il est vivement recommandé de convertir la base de données en InnoDB. À l'invite de commande MySQL, importer le script :

```
mysql> source /usr/share/doc/zarafa/db-convert-4.20-to-innodb.sql
```

En fonction de la taille de la base de données et de la vélocité du système, l'exécution de ce script peut prendre longtemps, voir très longtemps. Prévoir jusqu'à 8 heures pour effectuer cette conversion dans le cas d'une base de données d'une taille de plusieurs gigaoctets. Si les paramètres de mémoire MySQL ont été optimisés avant de lancer ce script, il s'exécutera beaucoup plus rapidement.

```
db-convert-4.2x-to-5.00
```

Ce script Perl met à jour le format 4.2x de la base de données vers le format 5.0. Ce script calcule et ajoute une colonne 'store' à la table de propriétés. Ceci permet d'ordonner la table sur le disque, accélérant ainsi le débit des données. Ce script s'exécute de la même façon que le script **`db-convert-4.1-to-4.2`**.

En fonction de la taille de la base de données et de la vélocité du système, l'exécution de ce script peut prendre un certain temps, probablement de l'ordre de 10 à 30 minutes sur une machine puissante.

Note

Il est recommandé d'exécuter ce script avec le logiciel Screen, afin qu'il puisse continuer son exécution en arrière-plan.

12.2. Mise à jour de la version 5.0 vers les versions 5.1x et supérieures

Le serveur Zarafa 5.10 peut lui-même mettre sa base de données à jour. C'est possible depuis la version de la base de données utilisée dans Zarafa 5.0. Lors d'une mise à jour depuis une version 4.x vers une version 5.10 ou supérieure, la mise à jour de la base de données vers le format 5.0, devra être effectuée à l'aide des scripts décrits ci-dessus. Le serveur 5.10 pourra ensuite être démarré et il finalisera automatiquement la mise à jour depuis la version 5.0 vers sa propre version 5.10.

Toutes les versions suivantes de Zarafa effectuent automatiquement la mise à jour de la base de données des versions 5.0 ou supérieures.

12.3. Changements notoires depuis les versions 4.x et 5.x

Une option de configuration du fichier **`server.cfg`** a été modifiée depuis la version 4.20. L'option **`server_name`** a été renommée **`server_bind`**. Un fichier de configuration contenant des erreurs dans les noms d'options ou faisant référence à des options inexistantes neutralisera le service et l'empêchera de démarrer. Toutes les erreurs trouvées dans le fichier de configuration seront affichées.

Dans la version 5.0, certaines options inutilisées ont été supprimées de la configuration du serveur. La gestion de SQLite a été retirée, et par conséquent, l'option **`internal_path`** a également disparue. Si cette option est toujours présente dans le fichier **`server.cfg`**, veuillez retirer cette ligne avant de lancer l'exécution de **`zarafa-server`**.

Les options qui ne sont pas définies dans le fichier de configuration conserveront leurs valeurs par défaut. Les valeurs par défaut se trouvent dans le fichier d'exemple de configuration situé dans **`/usr/share/doc/zarafa/example-config`**. Par ailleurs, la page du manuel se rapportant spécifiquement à ce service peut être consultée :

```
man zarafa-<service>.cfg
```

Les services Zarafa ne s'exécutaient pas en daemon dans les versions précédant la 5.0. Cependant, les versions 5.0 et supérieures se lancent en daemon, et s'exécutent en arrière-plan. Pour revenir au comportement antérieur, veuillez utiliser l'option **-F** d'un service afin de le garder au premier plan.

Les autres modifications de configuration se situent au niveau de la passerelle. Les valeurs par défaut affectant **ssl_private_file_key** et **ssl_certificate_file** ont été modifiées. Leur répertoire par défaut est désormais **/etc/zarafa/gateway/**, pour permettre la distinction avec les fichiers du service SSL.

Annexe B; description des attributs LDAP

Cette annexe détaille tous les attributs LDAP disponibles dans le schéma Zarafa. Le schéma Zarafa est disponible dans le kit d'intégration Active Directory ou dans le répertoire **`/usr/share/doc/zarafa`**.

Ne pas oublier que les fichiers de configuration LDAP de Zarafa sont très flexibles, et que ces attributs ne sont pas forcément tous utilisés.

zarafaQuotaOverride

Cet attribut est utilisé pour se substituer au quota par défaut, qui est configuré dans le fichier**`/etc/zarafa/server.cfg`**. Cet attribut doit toujours être activé pour pouvoir personnaliser les niveaux de quota.

OID	1.3.6.1.4.1.26278.1.1.1.1
Syntaxe	Nombre entier
Attribut unique ou multivalué	Attribut unique

zarafaQuotaWarn

Cet attribut contient le niveau du quota d'avertissement en Mo.

OID	1.3.6.1.4.1.26278.1.1.1.2
Syntaxe	Nombre entier
Attribut unique ou multivalué	Attribut unique

zarafaQuotaSoft

Cet attribut contient le niveau de quota modéré en Mo.

OID	1.3.6.1.4.1.26278.1.1.1.3
Syntaxe	Nombre entier
Attribut unique ou multivalué	Attribut unique

zarafaQuotaHard

Cet attribut contient le niveau de quota strict en Mo.

OID	1.3.6.1.4.1.26278.1.1.1.4
Syntaxe	Nombre entier
Attribut unique ou multivalué	Attribut unique

zarafaUserDefaultQuotaOverride

Cet attribut se substitue aux niveaux de quota du système pour tous les utilisateurs d'un tenant.

OID	1.3.6.1.4.1.26278.1.1.1.5
Syntaxe	Nombre entier

Attribut unique ou multivalué	Attribut unique

zarafaUserDefaultQuotaWarn

Cet attribut contient le niveau du quota d'avertissement en Mo pour tous les utilisateurs d'un tenant.

OID	1.3.6.1.4.1.26278.1.1.1.6
Syntaxe	Nombre entier
Attribut unique ou multivalué	Attribut unique

zarafaUserDefaultQuotaSoft

Cet attribut contient le niveau du quota modéré en Mo pour tous les utilisateurs d'un tenant.

OID	1.3.6.1.4.1.26278.1.1.1.7
Syntaxe	Nombre entier
Attribut unique ou multivalué	Attribut unique

zarafaUserDefaultQuotaHard

Cet attribut contient le niveau du quota strict en Mo pour tous les utilisateurs d'un tenant.

OID	1.3.6.1.4.1.26278.1.1.1.8
Syntaxe	Nombre entier
Attribut unique ou multivalué	Attribut unique

zarafaAdmin

cet attribut confère les droits d'administrateur à un utilisateur.

OID	1.3.6.1.4.1.26278.1.1.2.1
Syntaxe	Nombre entier
Attribut unique ou multivalué	Attribut unique

zarafaSharedStoreOnly

Cet attribut transformera une boîte aux lettres en base de stockage partagée. Il n'est pas possible de s'identifier sur une base de stockage partagée.

OID	1.3.6.1.4.1.26278.1.1.2.2
Syntaxe	Nombre entier
Attribut unique ou multivalué	Attribut unique

zarafaAccount

Cet attribut peut être utilisé avec les filtres de recherche LDAP afin de filtrer les utilisateurs et les groupes.

OID	1.3.6.1.4.1.26278.1.1.2.3
Syntaxe	Nombre entier
Attribut unique ou multivalué	Attribut unique

zarafaSendAsPrivilege

Cet attribut est conçu pour les utilisateurs et les groupes devant posséder les autorisations "envoyer en tant que" l'utilisateur lorsque cet attribut est ajouté.

OID	1.3.6.1.4.1.26278.1.1.2.4
Syntaxe	DN ou DirectoryString
Attribut unique ou multivalué	Multivalué

zarafaMrAccept

Cet attribut permet de configurer l'acception automatique des demandes de réunion. Cet attribut **`n'est pas`** utilisé dans les versions actuelles de Zarafa.

OID	1.3.6.1.4.1.26278.1.1.2.5
Syntaxe	Nombre entier
Attribut unique ou multivalué	Attribut unique

zarafaMrDeclineConflict

Cet attribut refusera les demandes de réunion lorsque l'agenda contient déjà des rendez-vous. Cet attribut **`n'est pas`** utilisé dans les versions actuelles de Zarafa.

OID	1.3.6.1.4.1.26278.1.1.2.6
Syntaxe	Nombre entier
Attribut unique ou multivalué	Attribut unique

zarafaMrDeclineRecurring

Cet attribut refusera des demandes de réunion lorsqu'elles sont définies comme récurrentes. Cet attribut **`n'est pas`** utilisé dans les versions actuelles de Zarafa.

OID	1.3.6.1.4.1.26278.1.1.2.7
Syntaxe	Nombre entier
Attribut unique ou multivalué	Attribut unique

zarafaId

Cet attribut peut être utilisé comme ID générique unique, par exemple pour les utilisateurs et les groupes. Par défaut, cet attribut **`n'est pas`** utilisé par Zarafa.

OID	1.3.6.1.4.1.26278.1.1.2.8
Syntaxe	Nombre entier
Attribut unique ou multivalué	Attribut unique

zarafaResourceType

Cet attribut configure le type de ressource d'une base partagée. Les options disponibles sont **`Pièce`** ou "Équipement"

OID	1.3.6.1.4.1.26278.1.1.2.9

Syntaxe	DirectoryString
Attribut unique ou multivalué	Attribut unique

zarafaResourceCapacity

Cet attribut numérotera les salles ou les équipements disponibles.

OID	1.3.6.1.4.1.26278.1.1.2.10
Syntaxe	Nombre entier
Attribut unique ou multivalué	Attribut unique

zarafaHidden

Cet attribut masquera l'objet dans le carnet d'adresse global. Ne pas oublier que les objets ne sont jamais caché des administrateurs.

OID	1.3.6.1.4.1.26278.1.1.2.11
Syntaxe	Nombre entier
Attribut unique ou multivalué	Attribut unique

zarafaEnabledFeatures

Contrôles dont les fonctionnalités sont explicitement activées pour un utilisateur, et qui se substituent à toutes les fonctionnalités ayant été désactivées dans la configuration du serveur.

OID	1.3.6.1.4.1.26278.1.1.2.13
Syntaxe	Chaîne
Attribut unique ou multivalué	Multivalué

zarafaDisabledFeatures

Contrôles dont les fonctionnalités sont explicitement désactivées pour un utilisateur.

OID	1.3.6.1.4.1.26278.1.1.2.14
Syntaxe	Chaîne
Attribut unique ou multivalué	Multivalué

zarafaAliases

Cet attribut contient tous les autres adresses électroniques et les alias de l'utilisateur.

OID	1.3.6.1.4.1.26278.1.1.3.1
Syntaxe	DirectoryString
Attribut unique ou multivalué	Multivalué

zarafaUserServer

Cet attribut sera le serveur principal de l'utilisateur dans un environnement multi-serveur.

OID	1.3.6.1.4.1.26278.1.1.4.1

Syntaxe	DirectoryString
Attribut unique ou multivalué	Attribut unique

zarafaSecurityGroup

Cet attribut spécifie si un groupe a des privilèges de sécurité. Lorsque l'attribut est défini avec la valeur `0`, le groupe sera considéré comme une liste de diffusion.

OID	1.3.6.1.4.1.26278.1.2.2.1
Syntaxe	Nombre entier
Attribut unique ou multivalué	Attribut unique

zarafaViewPrivilege

Cet attribut contient les tenant ayant des privilèges d'accès sur le tenant sélectionné.

OID	1.3.6.1.4.1.26278.1.3.2.4
Syntaxe	DirectoryString
Attribut unique ou multivalué	Multivalué

zarafaAdminPrivilege

Cet attribut est conçu pour les utilisateurs d'un autre tenant qui possèdent des droits d'administrateur sur le tenant sélectionné.

OID	1.3.6.1.4.1.26278.1.3.2.5
Syntaxe	DirectoryString
Attribut unique ou multivalué	Multivalué

zarafaSystemAdmin

Cet attribut permet de spécifier les utilisateurs qui possèdent des droits d'administrateur sur le tenant sélectionné.

OID	1.3.6.1.4.1.26278.1.3.2.6
Syntaxe	DirectoryString
Attribut unique ou multivalué	Multivalué

zarafaQuotaUserWarningRecipients

Cet attribut est conçu pour les utilisateurs qui recevront un courriel de notification lorsqu'un utilisateur a dépassé son quota.

OID	1.3.6.1.4.1.26278.1.3.1.5
Syntaxe	DirectoryString
Attribut unique ou multivalué	Multivalué

zarafaQuotaCompanyWarningRecipients

Cet attribut est conçu pour les adresses électroniques des utilisateurs qui recevront un courriel de notification lorsqu'un tenant a dépassé son quota.

OID	1.3.6.1.4.1.26278.1.3.1.6
Syntaxe	DirectoryString
Attribut unique ou multivalué	Multivalué

zarafaCompanyServer

Cet attribut est conçu pour le serveur principal du tenant dans un environnement multi-tenant.

OID	1.3.6.1.4.1.26278.1.3.4.1
Syntaxe	DirectoryString
Attribut unique ou multivalué	Attribut unique

zarafaHttpPort

Cet attribut est conçu pour le port dédié aux connexions HTTP dans un environnement multi-serveur.

OID	1.3.6.1.4.1.26278.1.4.4.1
Syntaxe	Nombre entier
Attribut unique ou multivalué	Attribut unique

zarafaSslPort

Cet attribut est conçu pour le port dédié aux connexions HTTPS dans un environnement multi-serveur.

OID	1.3.6.1.4.1.26278.1.4.4.2
Syntaxe	Nombre entier
Attribut unique ou multivalué	Attribut unique

zarafaFilePath

Cet attribut est conçu pour le socket UNIX ou le canal nommé du serveur dans un environnement multi-serveur.

OID	1.3.6.1.4.1.26278.1.4.4.3
Syntaxe	DirectoryString
Attribut unique ou multivalué	Attribut unique

zarafaContainsPublic

Cet attribut activera la base de stockage publique pour le nœud spécifié d'un environnement multi-serveur. S'assurer qu'un seul et unique nœud n'ait activé cet attribut.

OID	1.3.6.1.4.1.26278.1.4.4.4
Syntaxe	Nombre entier
Attribut unique ou multivalué	Attribut unique

zarafaFilter

Cet attribut est conçu pour le filtre LDAP devant être appliqué sur une liste d'adresses ou sur un groupe dynamique.

OID	1.3.6.1.4.1.26278.1.5.5.1
Syntaxe	DirectoryString
Attribut unique ou multivalué	Attribut unique

zarafaBase

Cet attribut est conçu pour la base de recherche LDAP devant être appliqué a une liste d'adresses ou à un groupe dynamique.

OID	1.3.6.1.4.1.26278.1.5.5.2
Syntaxe	DirectoryString
Attribut unique ou multivalué	Attribut unique

Appendix C: Example LDIF

The LDIF below shows an example of LDAP configuration for a single tenant setup.

```
dn: dc=example,dc=com
objectClass: dcObject
objectClass: organization
dc: zarafa
description: My LDAP Root
o: example.com

dn: cn=Manager,dc=example,dc=com
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
cn: Manager
userPassword: secret
description: LDAP administrator

dn: ou=Addresslists,dc=example,dc=com
objectClass: organizationalUnit
objectClass: top
ou: Addresslists

dn: ou=People,dc=example,dc=com
objectClass: organizationalUnit
objectClass: top
ou: People

dn: ou=Groups,dc=example,dc=com
objectClass: organizationalUnit
objectClass: top
ou: Groups

dn: ou=Contacts,dc=example,dc=com
objectClass: organizationalUnit
objectClass: top
ou: Contacts

dn: cn=Mary Poppins,ou=Contacts,dc=example,dc=com
objectClass: inetOrgPerson
objectClass: top
objectClass: zarafa-contact
uidNumber: 1001
sn: Poppins
cn: Mary Poppins
mail: mary@poppins.org

dn: uid=john,ou=People,dc=example,dc=com
objectClass: posixAccount
objectClass: top
objectClass: zarafa-user
objectClass: inetOrgPerson
gidNumber: 1000
cn: John Doe
homeDirectory: /home/john
mail: john@example.com
uidNumber: 1000
zarafaAliases: j.doe@example.com
zarafaUserServer: node1
uid: john
zarafaAccount: 1
zarafaAdmin: 0
sn: Doe
userPassword: john
```

```
zarafaQuotaOverride: 1
zarafaEnabledFeatures: imap
zarafaDisabledFeatures: pop3
zarafaQuotaWarn: 1000000000
zarafaQuotaSoft: 1100000000
zarafaQuotaHard: 1200000000

dn: cn=Example addresslist,ou=Addresslists,dc=example,dc=com
objectClass: zarafa-addresslist
objectClass: top
cn: Example addresslist
zarafaFilter: (mail=*@example.com)

dn: cn=Example security group,ou=Groups,dc=example,dc=com
objectClass: posixGroup
objectClass: top
objectClass: zarafa-group
zarafaHidden: 0
cn: Example security group
gidNumber: 1000
memberUid: john
zarafaAccount: 1
description: Example security group
zarafaSecurityGroup: 1

dn: cn=Example distribution group,ou=Groups,dc=example,dc=com
objectClass: posixGroup
objectClass: top
objectClass: zarafa-group
zarafaHidden: 0
cn: Example distribution group
memberUid: john
zarafaAccount: 1
gidNumber: 1001
description: Example distribution group
zarafaSecurityGroup: 0
```