
KC Administrator Manual

Release 8.0.0

Kopano BV

Feb 21, 2022

1	Abstract	2
2	Introduction	3
2.1	Intended Audience	3
2.2	Architecture	3
2.3	Components	4
2.4	Protocols and Connections	5
3	Installing	6
3.1	System Requirements	6
3.2	Installation	9
3.3	Troubleshooting Installation Issues	11
3.4	SSL	12
4	Upgrading	14
4.1	Preparing	14
4.2	Creating backups	15
4.3	KC 8 dependencies	16
4.4	Performing the Upgrade on RPM based distributions	16
4.5	Performing the Upgrade on Debian based distributions	16
4.6	Finalizing the upgrade	18
5	Configure KC Components	20
5.1	Configure the Kopano Server	20
5.2	Configure Kopano Konnect	27
5.3	Configure Kopano Kraph	30
5.4	Configure the Kopano Spooler	31
5.5	Configure Kopano Caldav	32
5.6	Configure Kopano Gateway (IMAP and POP3)	33
5.7	Configure Kopano Quota Manager	34
5.8	Configure Kopano Search	35
5.9	Configure Kopano WebApp	37
5.10	Configure Kopano for user management with LDAP (e.g. OpenLDAP/ADS)	38
5.11	Postfix integration	46
5.12	Configure Z-Push (ActiveSync for Mobile Devices)	51
6	Special KC Configurations	58
6.1	Running KC components beyond localhost	58
6.2	Multi-tenancy configurations	59
6.3	Multi-server setup	64
6.4	Single Instance Attachment Storage	69

6.5	Single Sign On with KC	70
6.6	Tracking messages with Kopano Archiver	77
6.7	Kopano Python plugin framework	78
6.8	Running KC multi-server behind Reverse Proxy	81
6.9	Running KC with Active Directory in multi-forest environment	84
6.10	Configuring kopano-spamd for automatic spam/ham learning	85
7	Managing KC Components	86
7.1	Starting the services	86
7.2	Logging options	87
7.3	Security logging	87
7.4	Kopano statistics monitoring	90
7.5	Soft Delete system	91
8	User Management	92
8.1	Public folder	92
8.2	General usage of kopano-admin tool	92
8.3	Users management with DB plugin	94
8.4	Users management with UNIX plugin	97
8.5	User Management with LDAP or Active Directory	98
8.6	LDAP Condition examples	103
8.7	Kopano Feature management	103
8.8	Resource configuration	105
8.9	Out of office management	106
9	Performance Tuning	108
9.1	Hardware Considerations	108
9.2	Memory Usage setup	109
9.3	Setup of modules on different servers	111
10	Backup & Restore	113
10.1	Softdelete restore	113
10.2	Full database dump	114
10.3	Brick-level backups	115
11	High Availability	116
11.1	High Availability example setups	116
11.2	Installing	118
11.3	DRBD device initialization	121
11.4	Pacemaker configuration	123
11.5	Testing configuration	126
11.6	Testing a node failure	127
11.7	Testing a resource failure	127
11.8	Getting more information	127
12	Release Notes	128
12.1	Release notes for 8.5.0 (2018-02-05)	128
12.2	Release notes for 8.4.7	129
12.3	Release notes for 8.4.6 (2018-02-02)	129
12.4	Release notes for 8.4.5 (2017-12-15)	129
12.5	Release notes for 8.4.4 (2017-11-23)	129
12.6	Release notes for 8.4.3 (2017-11-07)	130
12.7	Release notes for 8.4.2 (2017-11-02)	130
12.8	Release notes for 8.4.1 (2017-11-01)	130
12.9	Release notes for 8.4.0 (2017-10-30)	130
12.10	Release notes for 8.3.5 (unreleased/state of 2017-10-31)	132
12.11	Release notes for 8.3.4 (2017-09-01)	132
12.12	Release notes for 8.3.3 (2017-08-09)	133
12.13	Release notes for 8.3.2 [2017-07-06]	133

12.14	Release notes for 8.3.1 [2017-06-20]	133
12.15	Release notes for 8.3.0 [2017-04-27]	134
12.16	Release notes for 8.2.0 [2017-02-17]	135
12.17	Kopano Core 8.1.0	137
12.18	Kopano Core 8.0.1	137
13	Compiling from source	139
13.1	Installing Kopano Core from Source	139
13.2	Installing Kopano MMC Snap-in from Source	139
14	Appendix A: Upgrade strategies	143
14.1	Upgrade from Zarafa Collaboration Platform	143
15	Appendix B: LDAP attribute description	144
16	Appendix C: Example LDIF	150
17	Appendix D: Common MAPI Errors	152
18	Legal Notice	155

Edition 8.3.0 - Kopano Team

This document, the Kopano Core Administrator Manual, describes how to install, upgrade, configure and maintain KC on your system. In addition various advanced configurations and integration options are covered.

CHAPTER 1

Abstract

Kopano Core provides the core MAPI-enabled messaging stack with the stability and flexibility of the linux platform. Kopano Core acts as the solid foundation for groupware messaging based on MAPI and enabling rich web clients such as Kopano WebApp or Kopano DeskApp as well as mobile and sync clients. With the modular architecture, Kopano Core hereby allows a variety of setup scenarios, scalable from a very low powered system to a multi-datacenter setup providing messaging capabilities to tens of thousands of users.

Kopano Core is entirely open source, licensed under the [GNU Affero General Public License version 3](#), and can be downloaded from [Kopano's download servers](#).

Kopano Core is provided in two ways:

- Repositories, available to customers with a valid Kopano subscription.
- Downloadable & installable packages for community usage.

Kopano Groupware Core (KGC) is an open source software suite capable of providing a complete MAPI-based groupware stack with extensive interfacing capabilities. It's architecture is very modular, makes use of standards wherever possible, and integrates with common open source components.

This document explains how to perform the most common administrative tasks with KGC.

Important: Although we, Kopano, try our best to keep the information in this manual as accurate as possible, we reserve the right to modify this information at any time, without prior notice.

2.1 Intended Audience

This manual is intended for system administrators responsible for installing, maintaining, and supporting the KGC deployment. We assume readers of this manual will have a thorough understanding of:

- Linux system administration concepts and tasks
- Email communication standards
- Security concepts
- Directory services
- Database management

2.2 Architecture

In accord with the UNIX philosophy, KGC consists of components that each take care of a well defined task. See the *KGC Architecture Diagram* which describes the relationships between the components and the protocols used. This diagram describes a simple setup as used by most of our customers. Only the most commonly used components are shown in the diagram.

The top part of the diagram shows the clients: software appliances by which users access their data. Some of these appliances are desktop applications, some are mobile applications.

In between “The Internet” and the “Kopano Server”, the infrastructure components of Kopano (blue) and some common infrastructure components (grey) can be found. These components are needed to facilitate communication between the Kopano Server and various clients. Microsoft Outlook does not need any special infrastructure, but communicates directly with the Kopano Server using the ActiveSync protocol via Z-Push.

The Kopano Server is basically serving MAPI calls, while storing data in a MySQL database. For user authentication several methods are available (and discussed in this document), most common are servers that implement LDAP (e.g.: OpenLDAP, Microsoft Active Directory or any other LDAPv3 capable LDAP server).

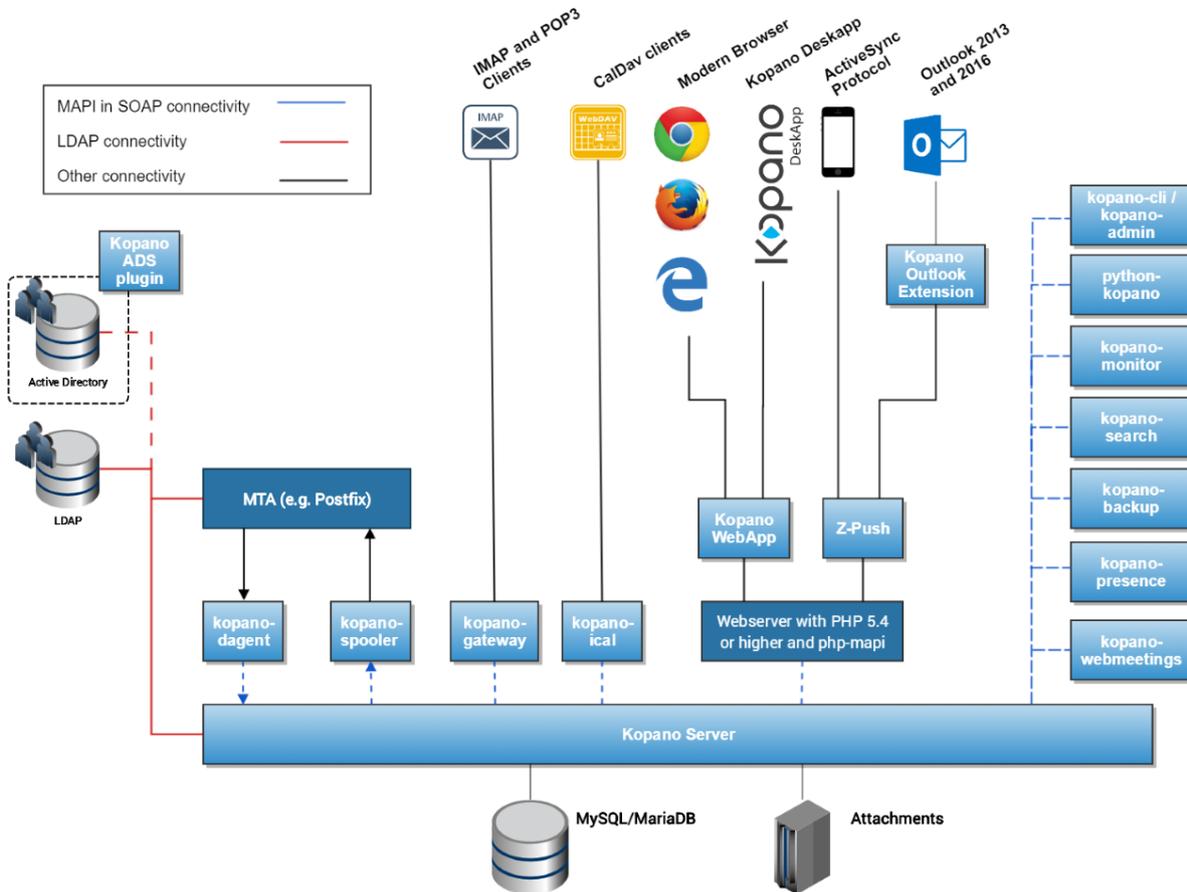


Figure 1.1. Kopano Suite Architecture Diagram

2.3 Components

Installations of Kopano Groupware may consist of the following components (list may not be complete):

- **Kopano Server** (`kopano-server`) - The server process accepts connections for all clients through SOAP (HTTP), and stores the data in an SQL database.
- **Kopano WebApp** (`kopano-webapp`) - The next generation collaboration web client, which offers integration with chat, presence and video conferencing.
- **Kopano Delivery Agent** and **Kopano Spooler** (`kopano-dagent`, `kopano-spooler`) - The tools which serve the email communication with the outside world. The dagent delivers mail from the Mail Transport Agent (MTA) to a Kopano user. The spooler sends mail waiting in the outgoing queue to the specified MTA.
- **Kopano Admin** (`kopano-admin`) - The command line administration tool is used to manage users, user information and groups.

- **Kopano Gateway** (`kopano-gateway`) - Optional service to provide POP3 and IMAP access to Kopano users.
- **Kopano Monitor** (`kopano-monitor`) - Service which monitors user stores for quota exceeds.
- **Kopano Caldav** (`kopano-caldav`) - Optional service that provides iCal and CalDAV support. CalDAV is recommended due to speed and less data transfer.
- **Kopano Backup** (`kopano-backup`) - A brick-level backup tool to create simple backups of users and stores with the ability to (partly) restore it at a later time.
- **Kopano Search** - Optional service to provide full text indexing. This offers fast searching through email and attachments.
- **Kopano Presence** - Kopano Presence Daemon which provides user presence to WebApp.
- **Webserver** - e.g. Apache, serves web pages of the WebApp to the users browser.
- **PHP** - The WebApp and Z-Push are written in this programming language.
- **PHP-MAPI extension** - Module for PHP to enable use of the MAPI layer. Through this module, MAPI functions are made accessible for PHP developers. This effectively means that MAPI web clients can be written. The WebApp is such a client.
- **Python-MAPI extension** - Module for Python to enable use of the MAPI layer. Through this module, MAPI functions are made accessible for Python developers.

For connectivity with mobile devices and Microsoft Outlook we recommend using [Z-Push](#) (see [Configure Z-Push \(ActiveSync for Mobile Devices\)](#)), an open-source implementation of the ActiveSync protocol.

2.4 Protocols and Connections

All applications which directly connect to the Kopano Server use MAPI in SOAP to do so (see the Architecture Diagram). Even the WebApp uses MAPI in SOAP (provided by the PHP-MAPI extension) to connect to the Kopano Server.

2.4.1 SOAP

SOAP is an abbreviation of Simple Object Access Protocol. It is a protocol to exchange data and make Remote Procedure Calls between applications over a network or Internet for that matter.

SOAP is based on XML and HTTP 1.1 (port 80, or port 443 in case of HTTPS). Because of these standards, it is possible to connect transparently through proxies, allowing connectivity over most networks without modifications.

2.4.2 Secure HTTP (HTTPS)

All Kopano clients using the SOAP protocol have the possibility to connect to the server over HTTP secured with SSL (HTTPS). All connections over the network will then be encrypted, making eavesdropping virtually impossible.

The Kopano Server must be configured to also accept SSL connections. By default, this is disabled, because it requires the creation of SSL certificates. When the server certificate is created, SSL connections can be directly accepted from a client. As an extra option, other Kopano components (like the Kopano Delivery Agent and the Kopano Spooler) can also connect over HTTPS to the server and authenticate using the Kopano Server's private key.

3.1 System Requirements

3.1.1 Hardware Recommendations

To give an estimate on the resource use of KC 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.

Table 2.1. Minimal Hardware Recommendations

Database Size / Users	CPU (Cores)	Memory	Harddisk	RAID level
< 5 GB / 1-25 users	2	2 GB	SATA, SAS, 7.2k	RAID 1
5 GB - 10 GB / 26-50 users	4	4 GB	SAS, 7.2k	RAID 1
10 GB - 20 GB / 51-100 users	4	6 GB	SAS, 10k	RAID 10
20 GB - 50 GB / 101-200 users	6	8 GB	SAS, 10k	RAID 10
50 GB - 100 GB / 201-300 users	6	10 GB	SAS, 10k	RAID 10
100 GB - 250 GB / 301-500 users	6	12 GB	SAS, 10k	RAID 10
> 250 GB / 501-1000 users	8	16 GB	SAS or SATA/SSD Hybrid, >= 10k	RAID 10

Important: Attachments do not require the same speed that is needed for the database storage. These can be safely put on slower disks/different RAID levels.

Important: Tuning of the server configuration and the individual software components for the specific onsite usage can drastically improve performance of your Kopano Core instance. For more than 500 users and/or a total mailbox storage bigger than 250 GB, as well as any high availability structures, it is advised to seek professional engineering support.

3.1.2 Connection/bandwidth Recommendation

In order to seamlessly connect Outlook clients to Kopano the network latency should not be higher than 20 ms. Network latencies of 200ms (500ms under exceptional circumstances) should not be exceeded in order to aid the user acceptance.

The needed bandwidth is very much dependent on the individual user behaviour. Based on large scale projects, we use the following key figures to calculate the minimal needed bandwidth:

For implementations with more than 100 users (with external access), we calculate with an average bandwidth utilization of “x (actual amount of users) * 8kbyte/s (ISDN speed)”. In real world scenarios not all users will require exactly the same amount of bandwidth at the exact same time, which still leaves room to serve short term higher demands of single users (like requesting an attachment from the server).

Given these key figures (with +20% TCP protocol overhead), the following minimum bandwidth for Outlook users can be calculated:

Table 2.2. Minimum bandwidth Requirements

Amount of users	Connection speed	Connection speed incl. TCP overhead
25	200 kbyte/s	240 kbyte/s
50	400 kbyte/s	480 kbyte/s
100	800 kbyte/s	960 kbyte/s
150	1200 kbyte/s	1440 kbyte/s
200	1600 kbyte/s	1920 kbyte/s
250	2000 kbyte/s	2400 kbyte/s
500	4000 kbyte/s	4800 kbyte/s
1000	8000 kbyte/s	9600 kbyte/s

Of course, these are only bare minimums and providing a higher bandwidth will increase download speeds.

3.1.3 Supported Platforms

KC consists of a large variety of components: some back-end components that are run on Linux platforms, and components that can be installed on the computers of end-users. In this section we list the different platforms that we support.

At the start of each general release cycle (like 7.x.x, 8.x.x or 9.x.x) we decide which platforms are supported. Usually that means the current release of that platform and the most recent previous release. During the major release cycle supported platforms can be added but not removed.

For a supported installation, make sure you use the x86_64 or 64-bit packages. Other architectures (e.g. i586, i686, ppc or arm) are not supported.

Table 2.3. Supported platforms for KC’s back-end components

OS Release	Supported CPU Architectures
Debian 8.x (Jessie) *	x86_64
Debian 9.x (Stretch)	x86_64
Debian 10.x (Buster)	x86_64
RHEL 6 *	x86_64
RHEL 7 *	x86_64
SLES 12 *	x86_64
Ubuntu 16.04 LTS (Xenial Xerus) *	x86_64
Ubuntu 18.04 LTS (Bionic Beaver)	x86_64
Ubuntu 20.04 LTS (Focal Fossa)	x86_64
Univention 4.3 *	x86_64
Univention 4.4	x86_64

Warning: Deprecation warning: distributions marked with a * will not be supported with Kopano Core 9.x and newer.

Important: Please be aware that this only specifies the architecture of the operating system and not the architecture of the client used.

These are the supported Microsoft Windows platforms for the components that require a Windows platform, namely: the ADS Plugin.

MS Windows Release	Supported CPU Architectures
Windows Server 2008 R2	64-bit
Windows Server 2012	64-bit
Windows Server 2016	64-bit
Windows Server 2019	64-bit
Windows 8	64-bit
Windows 10	64-bit

Important: Please note that Windows Server 2008 and Windows Server 2012 are no longer supported by Microsoft.

KC requires a system where glibc's functions (including semctl) work properly. Systems known to be problematic are for example old OpenVZ environments with kernel 2.6.x. This is for example due to the lack of /dev/shm being provided as tmpfs. Systems using semi-virtualization where glibc's default behavior is not maintained are not supported. This does not apply to fully or paravirtual solutions such as KVM, ESX, XEN, Hyper-V or any real isolated container format such as docker or lxc; These solutions are fully supported.

For more information about officially supported clients and support levels, please have a look at the [Support Lifecycle document](#).

3.1.4 Dependencies

In order to build or install KC back-end components, a number of requirements have to be met. These are the main dependencies of KC:

- **MySQL**, without an available MySQL Server the Kopano Server cannot run. There is no requirement to run MySQL Server on the same machine as the Kopano Server, therefore it is not a package dependency. MySQL version 4.0 or lower will not work correctly. KC is tested with the MySQL/MariaDB version provided by default by the supported distributions.
- **Apache** or any other webserver that supports PHP. KC is tested with Apache 2.2 and 2.4.
- **PHP**, standalone as CGI or as a webserver module. KC is tested with PHP >= 5.3 releases.
- **ICU** library that provides robust and full-featured Unicode and locale support.
- **SMTP** server of choice. KC is tested with Postfix, Exim, Sendmail and Qmail.
- **LDAP** server of choice (optional for user management). KC is tested with OpenLDAP, eDirectory and Microsoft Active Directory.
- **Catdoc** used to index text from Office documents.
- **Poppler-utils** used to index text from pdf files.
- **w3m** used to index HTML text from email.

Most of these dependencies are resolved automatically by the package manager of the Linux distribution that KC is being installed on. This allows the 3rd party components used by KC to be installed and upgraded automatically.

through the package manager of the distribution. Some dependencies in the table above are runtime dependencies, these have to be installed manually as they do not necessarily have to run on the same machine.

The default method of deploying KC is installing the packages on one of the Linux distributions we support, allowing the 3rd party components used by KC to be installed automatically through the package manager of the distribution. In this case the 3rd party components are upgraded in a standard way according to that distribution.

Note: If you're using Debian or Ubuntu and you're starting with a *fresh* install of your server, you can use **tasksel** to easily install the entire LAMP (Apache, MySQL, PHP) stack. This will provide all the packages which are required for the Kopano installation script to complete successfully.

3.2 Installation

There are multiple ways to install Kopano Core: (1) through a distribution's package manager and the package repositories provided by Kopano, (2) by manually installing the individual packages, and (3) from source. The following chapter gives an overview of how to install Kopano Core through repositories and individual packages.

When installing the provided packages, please always use the package provided for the individual distribution. Please see the distribution list in *Supported Platforms* for an overview of officially supported distributions. Packages available on the Kopano download server for distributions not mentioned in this table are provided as-is.

For per-distribution installation steps, please have a look at the [Kopano Knowledge Base](#).

Note: In an effort to have consistent behaviour between distributions Kopano packages do not start after installation or enable automatic restart after a reboot (usual behaviour for RHEL/SLES, but maybe not expected for Debian/Ubuntu).

3.2.1 Installing Kopano Core through the Kopano package repositories

To simplify the installation and updating of a Kopano system, Kopano provides packages repositories for customers with a valid subscription. An overview of the all available repositories can be found at <https://download.kopano.io/supported/>. The URL of each repository consists of the following parts:

- The base URL of the repository

This is always <https://download.kopano.io/supported/>. Navigation to this URL in a web browser gives an overview of all the available products.

- The product root.

This is the codename of the individual product, followed by a colon (:). For Kopano Core, the product root is "core:".

- The release type.

Can either be "final", "pre-final", or "master". Kopano products are provided in three different release types. The "master" release is a nightly build of the main development branch of the individual product. The "pre-final" release type includes pre release quality software, like beta or RC releases. The last type "final" included packages that have been released as stable releases.

For production systems, we recommend to use packages of the "final" release type, as only these are officially supported.

The last part is the identifier of the used distribution. The easiest way to check for the correct identifier of your distribution is to navigate to this URL in your browser. Additionally, you can also find a directory called "tarballs" at this location. This directory also contains tarballs of previous releases.

You can find more information in the [Kopano Knowledge Base](#).

After configuring the desired Kopano Core repository, Kopano can simply be installed through the meta package “kopano-server-packages”.

3.2.2 Manually Installing the individual Packages

Note: Do not mix packages of different distributions! Choose one distribution, and use only those packages. If this rule is not honored, errors will occur!

RPM-based distributions

Use the following command to install the KC packages on RPM-based distributions:

```
rpm -Uvh <package files you want to install>
```

Replace <package file> with the packages found in the tarball. Start with kopano-server-packages (in this order) then install the other packages. The package manager might find unresolved dependencies, try to install packages for these dependencies as normal would be done for that distribution (yum -i on Red Hat, zypper -i on SLES).

Note: Using distribution specific packaging tools, it may be easier to resolve package dependencies for RPM-based distributions. For SLES, you should use “zypper in <package>”, and for RHEL-based systems, “yum localinstall <package>”. If you have a subscription, we recommend the use of our package repositories.

DEB-based distributions

On DEB-based distributions (most commonly Debian and Ubuntu) use:

```
dpkg -i <package files you want to install>
```

Note: If you have a subscription, we recommend the use of our package repositories.

For the database, use:

```
apt install mysql-server  
# or depending on availability  
apt install mariadb-server
```

For Apache with the needed PHP support, use the following.

For PHP 5:

```
apt install apache2 libapache2-mod-php5
```

For PHP 7 (e.g. Ubuntu 16.04 and Debian 9):

```
apt install apache2 libapache2-mod-php7.0  
phpenmod kopano
```

If the Kopano packages fail to install because of dependencies, please use the following command to install these dependencies:

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

Note: The quickest way to install Kopano is not by selecting packages one by one to install and then resolving their dependencies, but by doing it the other way around. Therefore, it is recommended to simply remove packages you explicitly do not want (like *-dev and *-dbg) and simply installing the rest by issuing “dpkg -i *.deb” followed by “apt-get install -f” to get the missing dependencies from apt.

3.3 Troubleshooting Installation Issues

3.3.1 Server processes

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.

If errors when loading libraries occur or connecting to MySQL fails, the errors are printed in the log. Always check if the service was started correctly.

When an invalid configuration option is present in a configuration file, the service will not start. The wrong options will be printed on the console.

3.3.2 SELinux

If a distribution in combination with SELinux is used, an error message while logging in may appear when using WebApp. The default message suggests that the entered password is wrong or the Kopano server is not running. When SELinux is enabled, it is blocking the connection from the webserver to the Kopano server.

To create a minimal SELinux policy for Kopano, please walk through the following steps:

- Create a file kopano.te with the following lines:

```
module kopano 1.1;

require {
    type var_run_t;
    type postfix_postdrop_t;
    type httpd_t;
    type postfix_pipe_t;
    type initrc_t;
    class sock_file write;
    class unix_stream_socket connectto;
    class fifo_file { write getattr };
}

#===== httpd_t =====
allow httpd_t initrc_t:unix_stream_socket connectto;
allow httpd_t var_run_t:sock_file write;

#===== postfix_pipe_t =====
allow postfix_pipe_t initrc_t:unix_stream_socket connectto;
allow postfix_pipe_t var_run_t:sock_file write;

#===== postfix_postdrop_t =====
allow postfix_postdrop_t initrc_t:fifo_file { write getattr };
```

- Compile the SELinux policy with command: `checkmodule -M -m -o kopano.mod kopano.te`
- Packade the policy with command: `semodule_package -o kopano.pp -m kopano.mod`
- Copy the kopano.pp policy file to the directory: `/etc/selinux/targeted/modules/active/modules`
- Now load the policy with the command: `semodule -vi /etc/selinux/targeted/modules/active/modules/kopano.pp`

- To check if the policy is loaded, you can use `semodule -l`

Alternatively SELinux can be disabled by using the following command:

```
setenforce permissive
```

When it is chosen to disable SELinux, `/etc/sysconfig/selinux` also has to be edited, to disable it for after reboots too.

More SELinux information can be found on http://selinuxproject.org/page/Main_Page.

3.4 SSL

By default, the WebApp installation requires HTTPS to be configured, which is recommended. When SSL is not desired, it is possible to disable the configuration check for these security options inside the `config.php` file, and disable the option `CONFIG_CHECK_COOKIES_SSL`.

The following steps will guide you through the process of creating a self-signed certificate in order to secure Kopano WebApp. In environments where users are going to access WebApp and you do not want them to receive a warning message using a self-signed certificate, please follow the how to on requesting a certificate from an official certificate authority provider, which requires you to generate a CSR (certificate signing request) to get an officially signed certificate.

- Creating the directory to hold the certificate files:

```
mkdir /etc/apache2/certs
chmod 700 /etc/apache2/certs
cd /etc/apache2/certs
```

- Generating the key for the certificate. Follow the wizard and answer the questions required (as prompted) to generate the certificate.

```
openssl req -nodes -newkey rsa:2048 -keyout kopano-ssl.key -out kopano-ssl.csr
```

This creates two files. The file `kopano-ssl.key` contains a private key; do not disclose this file to anyone. Carefully protect the private key.

In particular, be sure to backup the private key, as there is no means to recover it should it be lost. The private key is used as input in the command to generate a Certificate Signing Request (CSR).

- You will now be asked to enter details to be entered into your CSR. What you are about to enter is what is called a Distinguished Name or a DN. For some fields there will be a default value, If you enter '.', the field will be left blank.

```
Country Name (2 letter code) [AU]: NL
State or Province Name (full name) [Some-State]: Zuid-Holland
Locality Name (eg, city) []: Delft
Organization Name (eg, company) [Internet Widgits Pty Ltd]: Kopano
Organizational Unit Name (eg, section) []: IT
Common Name (eg, YOUR name) []: example.kopano.com
Email Address []:
```

Use the name of the webserver as Common Name (CN). If the domain name (Common Name) is `domain.com` append the domain to the hostname (use the fully qualified domain name). The fields email address, optional company name and challenge password can be left blank for a webserver certificate.

- When ordering a certificate, you will need the contents of the `kopano-ssl.csr` file.

```
cat /etc/apache2/certs/kopano-ssl.csr
```

Paste the contents of the file into order form on the website you are ordering from. After receiving the certificate, follow the instructions given by your certificate reseller.

- Self-signing the certificate (Skip this step if you are purchasing a certificate)

```
openssl x509 -req -in kopano-ssl.csr -signkey kopano-ssl.key -out kopano-ssl.crt \  
-days 1825
```

4.1 Preparing

Before upgrading to a new version of KC, it is recommended to make a backup of the database and the configuration files, especially when upgrading is done by skipping some minor versions.

First stop the MTA server running on your server. Should there be any problems during the upgrade no e-mail will get lost, as mails would be queued for re-delivery until the MTA is back available. In case of postfix, run:

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

Now stop the running services, so the database is not in use anymore:

```
service kopano-spooler stop  
service kopano-server stop
```

And the optional services too, if they were started:

```
service kopano-dagent stop  
service kopano-gateway stop  
service kopano-ical stop  
service kopano-search stop  
service kopano-monitor stop
```

4.1.1 Database Attachments

With Kopano it is possible (and advised) to store all attachments outside the MySQL database. The split architecture of attachments was made to provide a more efficient MySQL caching system and get smaller, more manageable databases.

Configuring storage method

The attachment storage can be configured in the `server.cfg` file via the option `attachment_storage`. When using the install script the default value is set to `files`, so the attachments are stored in the filesystem per default. To store the attachment inside of the database, set this option to `database`. Make sure that there is enough space on the partition wherever your attachments are configured (whether in database or files).

Important: We do not recommend the usage of the “database” attachment store method. Using database driver is stable, but will make your database, depending on the amount of attachments harder to manage and at the same time the efficiency of mysql caching will drastically be lowered, resulting in lower performance of the overall system. For production environments, we only recommend to use either ‘files’ or ‘s3’ attachment storage.

Migrating database attachments to files

Existing installations that already have the attachments in the database have the possibility to migrated to the file storage. In the /usr/share/doc/kopano directory there is a script available called db-convert-attachments-to-files. This perl script will directly login to MySQL and dump the attachments to the specified directory. You have the run the script with the following options:

```
perl db-convert-attachments-to-files mysqluser mysqlpassword mysqldatabase \
  path_to_filesystem [delete]
```

The last delete option is optional. This option will delete the attachments from the database. The script can ran multiple times after each other, without having the attachments duplicated. To enable this settings you have to restart your Kopano-server one time with the option `--ignore-attachment-storage-conflict`.

```
kopano-server --ignore-attachment-storage-conflict
```

Storage layout

The attachments are not all stored in a single directory, but spread over 200 directories. Below the specified attachment_path there are 10 directories (0 - 9). Each directory has 20 subdirectories (0 - 19). Through the attachment id in the database, the exact location can be calculated via a fast algorithm. The attachment is in one of the directories with the id as filename. The stored attachment is the same as the attachment you will see in your email. With the file command you can see the probable type of the attachment.

4.2 Creating backups

Now create backups of the database and configuration files. Make a copy of the /etc/kopano directory, which contains the configuration files.

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

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

```
cp -r /var/lib/kopano/attachments /var/lib/kopano/attachments.bck
```

To backup the MySQL database a `mysqldump` can be executed:

```
mysqldump -p --single-transaction --routines kopano > kopano.sql
```

or the complete mysql data directory can be copied:

```
/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.

4.2.1 Consistent backups

Creating consistent backups between the database and the filesystem is possible by for example using snapshotting methods. There is no direct necessity to backup all data simultaneously, it is recommended to backup database first (with `--single-transaction` or any alike snapshotting method) and to backup attachments afterwards. This makes sure that in the worst case there are minor extra attachments instead of any missing.

4.3 KC 8 dependencies

After the backup is successfully created, the Kopano packages can be upgraded. There are some new dependencies that need to be resolved before the packages can be updated.

Table 3.1. KC 8 dependencies

Distribution	Dependencies
Debian 7	libboost-filesystem1.49.0 ¹ , libboost-system1.49.0 ¹ , libicu48, w3m
Debian 8	libboost-filesystem1.55.0 ¹ , libboost-system1.55.0 ¹ , libicu52, w3m
RHEL6	boost-filesystem ¹ , boost-system ¹ , libicu, w3m
RHEL7	boost-filesystem ¹ , boost-system ¹ , libicu, w3m
SLES11	libicu, w3m
SLES12	libicu, w3m
Ubuntu 12.04	libboost-filesystem1.46.1 ¹ , libboost-system1.46.1 ¹ , libicu48, w3m
Ubuntu 14.04	libboost-filesystem1.54.1 ¹ , libboost-system1.54.1 ¹ , libicu52, w3m
Ubuntu 16.04	libboost-filesystem1.58.1 ¹ , libboost-system1.58.1 ¹ , libicu55, w3m

4.4 Performing the Upgrade on RPM based distributions

After the backups have been created the upgrade can be performed similarly to how a package would be installed manually. For RPM based installations use the following command:

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

Note: Not necessarily all packages are required in your environment. Especially packages for clustering for example are required only for HA environments and are not required to be installed. We recommend only the installation/upgrade of packages that are really used.

After the new packages are installed, the example configuration files found in the `/usr/share/doc/kopano/example-config` directory can be checked for new configuration options. The new changes can also be found in the section *Release Notes*.

4.5 Performing the Upgrade on Debian based distributions

Unpack the tarball:

```
tar zxvf core-8.x.x-<revision>-<distribution>-<arch>.tar.gz
```

Install the new libvmime 0.9 that comes with Kopano:

```
dpkg -Bi libvmime0*
```

¹ Not needed anymore for releases after 8.3.0.

Install the python-mapi packages that comes with Kopano:

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

For Debian based installations run the following command to upgrade the KC installation:

```
dpkg -Bi <package name>
```

Depending on the set of packages you may have installed, this command may end with errors on the “kopano” packages. Due to the big split and renaming of packages some conflicts are not directly resolvable by “dpkg”. If you receive any errors during the upgrade of these packages, a second try installing these packages using:

```
dpkg -i <package name>
```

or run the following command:

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

which should resolve everything properly.

When prompted about changed kopano configuration files it depends greatly on you current situation what the best option is.

After the new packages are installed, the example configuration files found in the `/usr/share/doc/kopano/example-config` directory can be checked for new configuration options. The new changes can also be found in the [Kopano Changelog document](#).

For most people, upgrading is as easy as upgrading the packages. But before you restart the services you should manually update your configuration files and optionally make some changes to your LDAP (or Active Directory) server. Debian packages will automatically restart services. Some services will not correctly restart because of configuration options that change. You might see errors on your screen, but this is normal, and not destructive. All your data will still be present. The config files have changed quite a bit. Use the diff command to find the differences between your version of the config file and the version shipped with KC in `/usr/share/doc/kopano/example-config`. Most important are the `server.cfg` and `ldap.cfg` (in case you use LDAP or Active Directory) files.

To protect the server from deleting users a safe mode option is available in the `server.cfg`. Enabling this option will disable all delete and create actions of users and groups.

Add the following option in the `/etc/kopano/server.cfg` to enable safe mode:

```
user_safe_mode = yes
```

Check the server logfile after starting the Kopano Server for detection of user changes. If no users are recreated or deleted the configuration file is correct and `user_safe_mode` can safely be disabled.

Important: It’s strongly advised only to use the `safe_mode` during upgrade testing. When the upgrade has successfully completed, the `safe_mode` should be disabled. Running a production system with `safe_mode` enabled can result in performance issues and unexpected behaviour (like not creating stores for new users).

Note: When using LDAP or Active Directory (ADS), set applicable shared stores to ‘room’ or ‘equipment’ resource types to extend the addressbook. (for ADS a new plugin is required, for LDAP there is a new `kopano.schema` file required)

If you list the users with “kopano-admin -l” and you get an “Object not found” error, then please do an `ldapsearch` on the commandline with the `ldap_search_base` as search base. Most likely you will get an error “Size limit exceeded” with the `ldapsearch`.

If you are using ADS and the `ldapsearch` returns a “Size limit exceeded”, please increase the “MaxPageSize” policy. This value is the maximum number of results that ADS is allowed to return to a LDAP query.

When finished the migration, and before running in production again, we recommend the following checklist:

- Check if all users are there
- Check if all groups are there
- Check if companies are correct (if running multi company)
- Check if multi-server is working correctly (if running a multi server setup)
- Check if group memberships are correct
- Check if send-as permissions are correct
- Check if contacts are there
- Check if addresslists are working
- Check if dynamic-groups are working

4.6 Finalizing the upgrade

After the new configuration options have been checked, the services can be started again:

```
service kopano-server start
service kopano-spooler start
```

The optional services can also be started again:

```
service kopano-dagent start
service kopano-gateway start
service kopano-ical start
service kopano-search start
service kopano-monitor start
```

Important: Run `kopano-search-upgrade-findroots.py` if you upgrade from a Kopano Core version lower than 8.2.0. Without running it searching in shared mailboxes is not available.

Since upgrades usually include a changed `php-mapi` extension, the webserver has to be restarted as well:

```
service apache2 restart
```

or

```
service httpd restart
```

KC has a new improved IMAP/POP3 gateway. The new gateway offers better compatibility and higher performance by using additional information which is stored in the database and in the Kopano attachment directory. As this additional information will use more disk space and is only used when users are connecting over IMAP, the IMAP/POP3 features are by default disabled.

When users should have access to IMAP or POP3 this feature has to be manually enabled. Read more about enabling/disabling features in *Kopano Feature management*.

To generate for all existing messages an optimized IMAP version, the `optimize-imap.py` script is available. By executing this script for every existing email the envelope structure and body structure are stored in the database. Additionally the whole RFC822 message file is generated and stored gzip compressed in the attachment directory.

The script will only generate this data for the users who have IMAP and POP3 enabled.

To execute the script use the following command:

```
python /usr/share/doc/kopano-gateway/optimize-imap.py
```

To optimize one or more specific users use the following command: `python /usr/share/doc/kopano-gateway/optimize-imap.py <user1> <user2> <user3>`

Note: For new emails received on Kopano Core the optimized IMAP data is stored automatically when users have IMAP or POP3 enabled.

Configure KC Components

Most KC 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 *Managing KC Components*.

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

```
/etc/init.d/kopano-<component name> restart
# or
service kopano-<component name> restart
# or
systemctl restart kopano-<component name>
```

5.1 Configure the Kopano Server

If a component requires custom configuration, these can be done in a system-wide configuration file located below */etc/kopano/*. The default name for such a configuration file takes the name of the component and adds *.cfg*.

Each services allows to specify a custom location for the configuration as well. Please consult the man page for the appropriate syntax.

```
/etc/kopano/<component name>.cfg
# example for kopano-server
/etc/kopano/server.cfg
```

Annotated example configuration files can be found below */usr/share/doc/kopano/example-config/*.

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 kopano-server.cfg
```

If a config option is not present in the configuration file, the default setting will be assumed. For most setups these defaults will already be fine. In this chapter we only explain the basic configuration option of Kopano Server.

The Kopano 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 Kopano 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 Kopano server connects over the network to the MySQL database– allow it to connect from the IP-address from which the Kopano Server will connect.

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

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

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

```
GRANT ALTER, CREATE, CREATE ROUTINE, DELETE, DROP, INDEX, INSERT, LOCK TABLES, \
      SELECT, UPDATE ON kopano.* TO 'kopano'@'localhost' IDENTIFIED BY 'password';
```

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

5.1.1 Configure language for store creation

Note: The below instructions are valid for Kopano Groupware Core installations from 8.6.8 and newer. Older installations used `/etc/default/kopano` (Debian based) or `/etc/sysconfig/kopano` (rpm based).

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

```
/etc/kopano/admin.cfg
```

Add/Change the option `default_store_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.

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

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

On Debian based systems the locale needs to be enabled in `/etc/locale.gen`. The following command can be used to easily enable and generate the needed locales:

```
dpkg-reconfigure locales
```

In Debian distributions the following entry in `/etc/apache2/envvars` needs to be set to force the locale for Apache, else locale specific characters might not be displayed correctly in the WebApp.

```
## 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
```

5.1.2 User Authentication

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

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 existing infrastructure. The `ldap` plugin can optionally also hold the required configuration for a multi-server Kopano environment. Multi-server support is only supported in the Kopano Enterprise edition.

More information on managing users can be found in *User Management*.

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

Table 4.1. User plugin comparison

Feature	DB	Unix	LDAP
Create/delete/ modify users	yes	yes	yes
Set aliases	On MTA level	On MTA level	yes
Hide users	•	•	yes
Sendas permissions	yes	yes	yes
Sendas permissions of groups	•	•	yes
Security Groups	yes	yes	yes
Distribution groups	•	•	yes
Hide groups	•	•	yes
Dynamic groups	•	•	yes
Contacts support	•	•	yes
Multi-tenancy support	yes	•	yes
Addresslists support	•	•	yes
Multi-server support	•	•	yes

Important: Although multi-tenancy is already possible when using the DB plugin, we strongly suggest using an LDAP backend when planning to host multiple tenants within one installation.

The DB Authentication Plugin

This plugin uses the Kopano MySQL database to store user and group information. The `kopano-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 `kopano-admin` tool, see *User Management*.

The Unix Authentication Plugin

Important: This plugin is mostly delivered for backwards compatibility. Newer setups should either use the db plugin or the ldap plugin (recommended).

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 `kopano-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 Kopano, 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 `kopano-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/kopano/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 Kopano 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 kopano-unix.cfg
```

The LDAP Authentication Plugin

The LDAP plugin is used for coupling any LDAP compliant server with the Kopano 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 KC.

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

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

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

For more details about configuring the LDAP plugin see *User Management with LDAP or Active Directory*.

5.1.3 Autoresponder

KC 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 `kopano-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/kopano/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:

```
SENDDDB=${TMP:-/tmp}/kopano-vacation-$USER.db
```

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

```
SENDDDBTMP=${TMP:-/tmp}/kopano-vacation-$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 `kopano-dagent` manual page which describes how to use an alternate script (using the `-a` option).

5.1.4 Storing attachments outside the database

Since version 6.0 it is possible to save the attachments outside the database. KC 7.0.5 and higher will use the filesystem as default location for attachment storage. For better database performance it is recommended to store attachments outside of the database.

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/kopano/server.cfg`.

```
attachment_storage = files
attachment_path = /var/lib/kopano/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/kopano`, and is named `db-convert-attachments-to-files`. This script can be used as follows:

```
db-convert-attachments-to-files <myuser> <mypass> <mydb> <dest path> [delete]
```

Note: The script can be executed while the `kopano-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 KC can be looked up the with the following MySQL statements:

```
use kopano;
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 `kopano-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.
- Make use of deduplication techniques (for example filesystem capabilities or through hardlinking) to further reduce hard disk space.

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.

Important: When using NFS as storage backend for Attachment-Store or as WebApp `TMP_PATH` we recommend turning of NFS locking by using the `-o nolock` mount option as this potentially can cause severe performance penalties.

5.1.5 SSL connections and certificates

The Kopano 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 Kopano Server.

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

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

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

```
mkdir /etc/kopano/ssl
chmod 700 /etc/kopano/ssl
```

If Kopano 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/kopano` 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.

Note: With the release of Kopano 8.2.0 all Kopano components now check the validity of the CN and subjectAltNames fields of the certificate. Unlike browsers Kopano does not stop at self-signed certificates, so these can be used as well.

```
cd /etc/kopano/ssl
sh /usr/share/doc/kopano/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. The Common Name has to reflect the hostname clients will use to connect to.

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_listen_tls = *: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/kopano/ssl/demoCA/cacert.pem
server_ssl_key_file = /etc/kopano/ssl/server.pem
server_ssl_key_pass = <password>
```

Restart the `kopano-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.

5.2 Configure Kopano Konnect

Kopano Konnect is an OpenID provider (OP) that directly integrates a web login and consent form. It brings support for both OpenID Connect (OIDC) and Open Authentication (OAuth 2.0). In addition to the easier integration with third-party applications, Kopano Konnect will also provide the authentication part for the Kopano RestAPI and clients consuming it.

Konnect was primarily designed to allow easy sign in with your Kopano account, but it is also able to validate users against an LDAP and a cookie-based backend. The architecture of Kopano Konnect favours a mostly stateless setup, where no session data is stored locally but rather encapsulated within the token (jwt) that is sent to the user. In fact, the only requirements that Konnect has apart from the user backend are an encryption secret key, a private key to sign the user tokens and if services are running under a different hostname a yaml file that services as a client registry. If the encryption secret key or private key are missing Konnect will generate a random key at startup (user sessions won't survive a restart in this case). For convinience these keys are auto-generated through systemd before Kopano Konnect starts.

5.2.1 Configuration

Configuration of Konnect is performed in `/etc/kopano/konnectd.cfg`. When running with Kopano, the only setting that needs changing is the OpenID Connect Issuer Identifier, which has to be set to the hostname Konnect is accessible from (ideally using the same hostname as the other Kopano apps).

```
# OpenID Connect Issuer Identifier.
# This setting defines the OpenID Connect Issuer Identifier to be provided by
# this Konnect server. Setting this is mandatory and the setting must be a
# https URL which can be accessed by all applications and users which are to
# use this Konnect for sign-in or validation. Defaults to "https://localhost" to
# allow unconfigured startup.
#oidc_issuer_identifier=https://localhost
```

Remember to restart Konnect by executing `systemctl restart kopano-konnectd` after making configuration changes.

5.2.2 Configure a Webserver for Konnect

Kopano Konnect needs to be accessible on certain defined paths to fully work. This chapter gives example configuration snippets for Apache and Nginx. After the below snippets have been added your should be able to open `https://your-domain/signin/v1/welcome` and be greeted with the Kopano login mask.

Apache

Put the following snippet into `/etc/apache2/conf-available` (or your local equivalent) and enable it:

```
ProxyPass /.well-known/openid-configuration http://localhost:8777/.well-known/
↪openid-configuration retry=0
ProxyPass /konnect/v1/jwks.json http://localhost:8777/konnect/v1/jwks.json retry=0
ProxyPass /konnect/v1/token http://localhost:8777/konnect/v1/token retry=0
ProxyPass /konnect/v1/userinfo http://localhost:8777/konnect/v1/userinfo retry=0
ProxyPass /konnect/v1/static http://localhost:8777/konnect/v1/static retry=0
ProxyPass /konnect/v1/session http://localhost:8777//konnect/v1/session retry=0

# Kopano Konnect login area
ProxyPass /signin/ http://localhost:8777/signin/ retry=0
```

Nginx

Put the following snippet into `/etc/nginx/sites-enabled/default` (or your local equivalent) and enable it:

```

upstream konnect {
    server 127.0.0.1:8777;
}

location /.well-known/openid-configuration {
    proxy_pass http://konnect/.well-known/openid-configuration;
}

location /konnect/v1/jwks.json {
    proxy_pass http://konnect/konnect/v1/jwks.json;
}

location /konnect/v1/token {
    proxy_pass http://konnect/konnect/v1/token ;
}

location /konnect/v1/userinfo {
    proxy_pass http://konnect/konnect/v1/userinfo;
}

location /konnect/v1/static {
    proxy_pass http://konnect/konnect/v1/static;
}

location /konnect/v1/session {
    proxy_pass http://konnect/konnect/v1/session;
}

location /signin/ {
    proxy_set_header Host $host;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_set_header X-Forwarded-Port $server_port;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_pass http://konnect/signin/;
}

```

5.2.3 Configure 3rd Party Applications to Authenticate using Konnect

Note: Want to share how to configure your application for Kopano Konnect? Just drop us an email to feedback (at) kopano (dot) io and we will include it in this section. Need help configuring a certain application for Kopano Konnect? Either open a topic on [our forum](#) or when using a subscription [open up a support case](#).

An important resource for configuring third-party clients for the use of Kopano Konnect is the so-called “discovery document”. When running the above proxy server configuration the discovery document will be available from `https://$(hostname)/.well-known/openid-configuration`. Some client applications are able to autoconfigure themselves from this url and for applications that do not support this, you will still be able to find a listing of all relevant url endpoints of Konnect and the supported claims and scopes.

Most claims and scopes supported by Konnect are defined by the [OpenID Connect standard](#), but there are also some additional scopes defined by us that I want to quickly introduce.

- `konnect/id` → will change the return value for the user id to a numerical value.
- `konnect/hashed_sub` (introduced in 0.8.0) → the subject is normally equal to the unique user id in Kopano, but this value can include characters that are not allowed on the client side (like the plus sign for example), we, therefore, have introduced an additional claim that will convert the subject to an url safe value for these applications.
- `kopano/gc` (introduced in 0.9.0) → a scope that will be used by the by the Kopano Rest API to determine the unique user id, if `konnect/hashed_sub` is used.

In case an application is running on the same domain name as Konnect it will be automatically trusted. In case the application uses a different hostname, then the application needs to be “registered” before it can actually be used. For this registration, the file `identifier-registration.yaml` is used (please check the git repository for an [example file](#)).

Using Kopano Konnect to sign into Nextcloud

With this knowledge, we can now configure Nextcloud so that users will be able to sign into Nextcloud from Kopano Konnect. The below steps require the use of Kopano Konnect $\geq 0.8.0$ and have been tested with the Nextcloud 13.0.4 and the Nextcloud “Social Login” app in version 1.9.2 and 1.9.4.

The ability to sign into Nextcloud via OpenID Connect is unfortunately not a core component of Nextcloud, but instead needs to be achieved via the installation of a third party plugin (through the Nextcloud App Market). After the plugin has been installed administrators will see a new menu option called “Social Login” in the Nextcloud settings. From there a new “Custom OpenID Connect” can be configured.

The following values need to be filled in:

- Internal name → internal identifier can, for example, be set to “Kopano”
- Title → this is what will be displayed to the user on the Nextcloud login screen. Should be named “Kopano Konnect” or something else that the user can easily identify.
- Authorize url → this setting is named “authorization_endpoint” in the discovery document. value is for example `https://$(hostname)/signin/v1/identifier/_/authorize` (replace `$(hostname)` with your actual hostname here and in the following instances)
- Token url → this setting is named “token_endpoint” in the discovery document. Value is for example `https://$(hostname)/konnect/v1/token`
- User info URL (optional) → this setting is named “userinfo_endpoint” in the discovery document. Value is for example `https://$(hostname)/konnect/v1/userinfo`
- Client Id → self-defined value used to identify if the client registry is used (more below)
- Client Secret → self-defined value
- Scope → the list of required scopes, here `openid profile email konnect/hashed_sub` need to be entered.

In case Nextcloud is running on a different host or just on a different domain name, we also need to add an entry for it to the client registry. For this the following values need to be added to the end of the example file:

```
- id: nextcloud # same id as before
  name: Nextcloud # self-defined value
  application_type: web
  redirect_uris:
  - https://nextcloud.$(hostname)/apps/sociallogin/custom_oidc/Kopano
```

In the `redirect_uris` section the last part of the url (so the “Kopano”) needs to be the same value as the “internal name” in the Social Login configuration.

Once these settings are set there is a new button on the Nextcloud login page (with the text “Kopano Konnect”) that users can use to sign into Nextcloud.

In case the Kopano users already exist inside of Nextcloud (since both systems use a common ldap tree for example) it is recommended to check the options “disable auto create new users” and “allow users to connect social logins with their account” inside of the Social Login settings to prevent users from accidentally creating new user accounts when trying to login.

Once this is done the user can then link his existing Nextcloud user to the OpenID Connect user in his personal settings below the item “additional settings”.

Using Kopano Konnect to sign into services supporting ID4me

Starting with version 0.21.0 Kopano Konnect can be used to login into services that support logging in via ID4me. These services are commonly referred to as a “Relying Party”. The following requirements have to be met to use Kopano Konnect as a “Identity Agent” within the ID4me specifications:

- A domain name that is configured with DNSSEC
- A dns TXT record with the name `_openid` and the value `v=OID1;iss=kopano.dev;clp=kopano.dev` (where `kopano.dev` needs to be replaced with the actual FQDN Konnect can be reached at)
- A valid/trusted SSL certificate for the system providing Konnect
- The system running Konnect needs to be publicly accessible

When the above requirements are met, the only configuration change within Konnect is to allow dynamic client registration by setting `allow_dynamic_client_registration = yes` in `konnectd.cfg`.

5.3 Configure Kopano Kraph

Kopano API provides a REST web service with the endpoints to interface with Kopano via HTTP APIs. In addition to installing `kopano-kapid`, Kopano API also needs to have the required REST endpoints, for Kopano Groupware this is provided by `kopano-grapi`. Kopano API is only required if the next generation clients such as Kopano Meet should be used and is not required if only Kopano WebApp and/or Z-Push should be used.

Note: Kopano API and Grapi are currently only supported on Debian 9, Ubuntu 16.04 and Ubuntu 18.04. For the upcoming Kopano Groupware Core 9.0 release we want to add the rest of our supported platforms.

5.3.1 Configuration

Configuration of Kapi is performed in `/etc/kopano/kapid.cfg`. When running with Kopano, the only setting that needs changing is the OpenID Connect Issuer Identifier, which has to be set to the hostname Konnect is accessible from.

```
# OpenID Connect Issuer Identifier.
oidc_issuer_identifier=https://localhost
```

Remember to restart Kapid by executing `systemctl restart kopano-kapid` after making configuration changes.

The use Kopano Api, some settings need to be adjusted in `kopano-server` as well

```
#####
# OPENID CONNECT SETTINGS

# Enable OpenID Connect Issuer Identifier
# When set, the server attempts OIDC discovery using the configured issuer
# identifier on startup. An Issuer Identifier is a case sensitive URL using the
# https scheme that contains scheme, host, and optionally, port number and path
# components. This no Issuer Identifier is set, OIDC support is disabled.
#kcoidc_issuer_identifier =

# Disable TLS validation for OpenID Connect requests
# When set to yes, TLS certificate validation is skipped for all requests
# related to OpenID connect. This is insecure and should not be used in
# production setups.
#kcoidc_insecure_skip_verify = no

# Timeout in seconds when to give up OpenID Connect discovery
```

```
# When the OIDC initialize timeout is reached, server continues startup without
# OIDC and all OIDC validation will fail until the discovery completes. When
# set to 0, the server startup does not wait for OIDC discovery at all.
#kcoidc_initialize_timeout = 60

...

# Set to 'yes' if you have Kerberos, NTLM or OpenID Connect correctly configured,
↪for single sign-on
#enable_sso = no
```

The option `kcoidc_issuer_identifier` has to be set to the hostname Konnect is accessible from. Once this has been done `enable_sso` can be set to `yes`.

5.3.2 Configure a Webserver for Kopano Kraph

Kopano Api needs to be accessible on certain defined paths to fully work. This chapter gives example configuration snippets for Apache and Nginx. After the below snippets have been added the API will be available from `https://your-domain/api/gc/`.

Apache

Put the following snippet into `/etc/apache2/conf-available` (or your local equivalent) and enable it:

```
ProxyPass /api/gc/ http://localhost:8039/api/gc/ retry=0
ProxyPass /api/pubs/ http://localhost:8039/api/pubs/ retry=0
```

Nginx

Put the following snippet into `/etc/nginx/sites-enabled/default` (or your local equivalent) and enable it:

```
upstream kapi {
    server 127.0.0.1:8039;
}

location /api/gc/ {
    proxy_pass http://kapi/api/gc/;
}

# kapi pubs
location /api/pubs/ {
    proxy_pass http://kapi/api/pubs;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "Upgrade";
}
```

5.4 Configure the Kopano Spooler

The Kopano-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 WebApp, the message is placed in the Outbox folder, and a submit message is sent to the Kopano server. The server notifies the Kopano 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.

5.4.1 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 Kopano 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 Kopano 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/kopano/server.sock`.

```
[logging]
```

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

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

```
man kopano-spooler.cfg
```

5.5 Configure Kopano Caldav

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

For an overview of all the configuration options of `kopano-ical`, use:

```
man kopano-ical.cfg
```

5.5.1 SSL/TLS for CalDAV

As mentioned before the Kopano 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 Kopano 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/kopano/privkey.pem 2048
```

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

```
openssl req -new -x509 -key /etc/kopano/privkey.pem -out /etc/kopano/cert.pem \
-days 1825
```

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. The same approach can be used to include any necessary intermediate certificates into the file.

5.6 Configure Kopano Gateway (IMAP and POP3)

The Kopano IMAP & POP3 Gateway enables users to view mail stored on the Kopano 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 Kopano Gateway itself connects to the Kopano 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 Kopano WebApp.

For an overview of all the configuration options of `kopano-gateway`, use:

```
man kopano-gateway.cfg
```

5.6.1 SSL/TLS for Gateway

The Kopano Gateway supports SSL/TLS using the OpenSSL library. For more information see [SSL/TLS for CalDAV](#), as the options are exactly the same for these two components.

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 WebApp). 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 Kopano 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 Kopano WebApp. 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.

5.7 Configure Kopano Quota Manager

Users can collect a lot of email, while disk space can be limited. The Kopano Quota Manager can be used to set server-wide or user specific space quotas. The Kopano 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.

5.7.1 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.

5.7.2 Setup quota per user

By using the `kopano-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.

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

Note: Set user quota with `kopano-admin` is not supported for the LDAP backend. With LDAP the properties are stored in the LDAP server per user. See the *User Management* for more information.

5.7.3 Monitoring for quota exceeding

The `kopano-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 `kopano-admin` when using the db or unix plugin, or by editing the LDAP values as described in the User Management section.

To start the `kopano-monitor`, use:

```
systemctl start kopano-monitor.service
```

or

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

The `kopano-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 kopano-monitor.cfg
```

5.7.4 Quota warning templates

When working with the `kopano-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`
- `companyquota_warning_template`

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

- `KOPANO_QUOTA_NAME` - The name of the user or company who exceeded his quota
- `KOPANO_QUOTA_COMPANY` - The name of the company to which the user belongs
- `KOPANO_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.
- `KOPANO_QUOTA_WARN_SIZE` - The quota warning 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.

5.8 Configure Kopano Search

The `kopano-search` service offers full text searching capabilities for the Kopano Server. The service will continuously index all mails of a single `kopano-server` instance. Each `kopano-server` instance in a multi-server setup needs its own `kopano-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).

5.8.1 Enabling the search service

To start the indexing service execute the following command:

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

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

```
search_enabled = yes
```

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

```
search_socket = file:///var/run/kopano/search.sock
```

Note: `Kopano-search` will only create the `search_socket` once the initial index has been created. Until this process is done `kopano-server` will complain about the socket not being accessible and use the direct database search instead.

5.8.2 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/kopano/search.cfg`:

```
index_path = /var/lib/kopano/search/
```

In this folder a file will be created for each store located on the Kopano 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 `kopano-search` must be stopped first before deleting the file for that particular store.

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

```
enable_enhanced_ics = yes
```

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

5.8.3 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/kopano/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/kopano/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.

5.9 Configure Kopano WebApp

For configuration instructions for Kopano WebApp please check the [WebApp Admin Manual](#).

5.9.1 Configure the Webserver

For instructions how to configure your webserver for Kopano WebApp please check the [WebApp Admin Manual](#).

Installing php-mapi

Client applications such as Kopano WebApp and Z-Push rely on the `php-mapi` module to access data stored in Kopano. The `php-mapi` packages is part of the Kopano Groupware Core installation packages. As PHP modules have to be compiled against the abi of the same PHP version it is later used with, Kopano provides packages matching the default PHP version for each of the supported distributions.

As a HTTP Reverse Proxy for Kopano Server

The transmitted data between the different client applications (for example Kopano WebApp) and `kopano-server` is compressed XML, wrapped in HTTP packets. The use of HTTP allows packets to be forwarded by a proxy (or a webserver with built-in proxy functionality). Please make sure that the proxy fully supports HTTP/1.1 and “Chunked Encoding” is available as a transport.

The following lines are an example of how Apache can be configured to forward incoming connections on port 80 to the Kopano Server on port 236. In case the Apache server also accepts HTTPS connections, the proxied connections can also be encrypted. The `proxy` and `proxy_html` modules of Apache need to be loaded for this to work (for example with `a2enmod proxy proxy_http`).

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

This means that URLs that begin with `/kopano` will be forwarded to `localhost` on port 236, where the Kopano Server listens for incoming connections. These lines can be placed globally, or within a `VirtualHost` declaration.

Note: Keep in mind that using a HTTP proxy will create some performance overhead on your system, so it is not recommended to use this for larger setups.

Note: Chunked encoding can be forced within Apache by setting `SetEnv proxy-sendchunked 1`.

Apache 2.2 is known to have some trouble with chunked encoding. Therefore we recommend using Apache 2.4 (or even Nginx) when planning to implement such a proxy.

5.10 Configure Kopano for user management with LDAP (e.g. OpenLDAP/ADS)

In several network environments an LDAP tree is used to keep track of various bits of information, most notably: users and their credentials. Kopano integrates with any LDAP server and directly supports the use of OpenLDAP and Microsoft ActiveDirectory (ADS).

As Kopano doesn't bundle a LDAP server of its own, this has to be setup separately if there is not yet a server available in the environment. Please read the documentation of the used Linux distribution on how to setup an LDAP server. Kopano provides an example LDIF file in *Appendix C: Example LDIF*.

Connections to the LDAP server usually run over port 389 or 636 (TLS/SSL). For best speed and reliability it is always recommended to install an LDAP server on the same host as the `kopano-server` itself. If required this local server can also be setup to replicate the main LDAP server. Besides performance improvements this also allows the `kopano-server` to function even when the main LDAP server is not available.

The following sections can be summarised into the following:

- Once the desired ldap backend is prepared, the file `/usr/share/doc/kopano/example-config/ldap.cfg` needs to be copied to `/etc/kopano/ldap.cfg`.
- Only files in `/etc/kopano` should be modified to configure Kopano or the ldap integration of Kopano. No files below `/usr/share` need to be modified!
- In `/etc/kopano/ldap.cfg` the include statement should be switched according to used type of ldap (OpenLDAP or ADS).
- Additionally the admin needs to define the required connection details in `/etc/kopano/ldap.cfg`.
- If there is a want/need to override any of the default configuration option (like for example `ldap_user_search_filter`), then this should be added to the bottom of `/etc/kopano/ldap.cfg`.

5.10.1 Configuring OpenLDAP to use the Kopano schema

To make managing Kopano users easier it is recommended to import the Kopano LDAP schema. The schema can be imported by issuing the following command:

```
zcat /usr/share/doc/kopano/kopano.ldif.gz | ldapadd -H ldapi:/// -Y EXTERNAL
```

5.10.2 Configure LDAP indices in OpenLDAP

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

Depending on the Kopano ldap configuration the attributes may be different.

Please check the OpenLDAP or syslog logfiles for further attributes which are not yet indexed and could be included to increase performance. Check below for an example log message:

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

When using the cn=config backend the following ldif file can be used to add the given attributes to the index of OpenLDAP:

```
dn: olcDatabase={1}mdb,cn=config
changetype: modify
replace: olcDbIndex
olcDbIndex: memberOf eq
olcDbIndex: entryCSN eq
olcDbIndex: entryUUID eq
olcDbIndex: objectClass eq
olcDbIndex: cn pres,eq,sub
olcDbIndex: gidNumber eq
olcDbIndex: mail pres,eq,sub
olcDbIndex: memberUid eq
olcDbIndex: ou eq
olcDbIndex: uid eq
olcDbIndex: uidNumber eq
olcDbIndex: uniqueMember eq
olcDbIndex: kopanoAccount eq,pres
olcDbIndex: kopanoAliases eq
olcDbIndex: kopanoViewPrivilege eq
olcDbIndex: sn pres,eq,sub
olcDbIndex: givenName pres,eq,sub
```

To import this the following command can be used:

```
cat optimize-index.ldif | ldapmodify -Y EXTERNAL -H ldapi:///
```

5.10.3 Configuring ADS to use the Kopano schema

Kopano Groupware Core provides an installer for extending the Active Directory schema and installing an Active Directory snap-in for managing the Kopano specific attributes.

With Kopano ADS Extension it is possible to create and modify the following objects in Active Directory:

- Kopano Users
- Kopano Groups
- Kopano Addresslists
- Kopano Dynamic Distribution Lists
- Kopano Computers (For usage with Multi-Server support)
- Kopano Companies (For usage with Multi-Company support)

Make sure you have prepared the AD role based on the usual best practice. We do not cover details of the setup, just a basic setup howto to get you started with Active Directory.

Before you deploy the AD role, you should make sure the following tasks have been completed:

- The administrator account has a strong password set
- The networking has been setup accordingly (static IP)
- The latest windows updates have been installed
- The user the installer is run with, should be part of the “Schema Admins” group

To make your system ready to provide an Active Directory, please follow the following steps. The steps are based on Windows Server 2016, Installation might differ slightly for previous versions.

1. In Server manager, select “Add roles and features” which starts the “Add Roles and Features Wizard”.
2. After confirming the “Before you begin” page as a reminder to the tasks also mentioned above, continue with “Next”.
3. Select “Role-based or feature-based installation”, continue with “Next”.
4. Select the Server you want to install the AD role(s) to, continue with “Next”.
5. Select “Active Directory Domain Services” and confirm the pop-up dialog to add the missing dependencies (including the management tools by keeping the checkmark enabled)
6. Select “Active Directory Lightweight Directory Services” and confirm the pop-up dialog to add the missing dependencies (including the management tools by keeping the checkmark enabled)
7. Continue with “Next” after having selected the roles mentioned above.
8. Continue with “Next” without selecting any additional (required) features.
9. Continue with “Next” confirming the installation of “AD DS” role.
10. Continue with “Next” confirming the installation of “AD LDS” role.
11. Continue with “Install” at the installation page to make the roles available. The checkmark for automatic restart can be set - in any case after deployment of AD a restart is required, whether this is done manually or automatic.

After these steps, with a reboot of the System your Active Directory should available and ready for installation of Kopano AD Extension.

The Kopano ADS Extension should be installed as a local administrator user on the Active Directory server which is the schema master. Following the above statement installs the schema role automatically on the system.

Note: Please restart the GUI after install of the Kopano ADS plugin to show the Kopano tab in the user details.

The Kopano ADS Extension is supported with Windows Server releases from 2008 R2 and 2016. The installation and schema extension are straight forward and it is possible to step through the setup by clicking the next button. The same installer can be used to install the MMC extension for non-AD-controllers to allow management of Kopano objects also remotely.

If the Kopano ADS Extension is installed, it is possible to edit the Kopano specific attributes. For editing a user go to *Active Directory Users and Computers*, select a user and get the properties. The Kopano tab should be available if the installation is successfully completed.

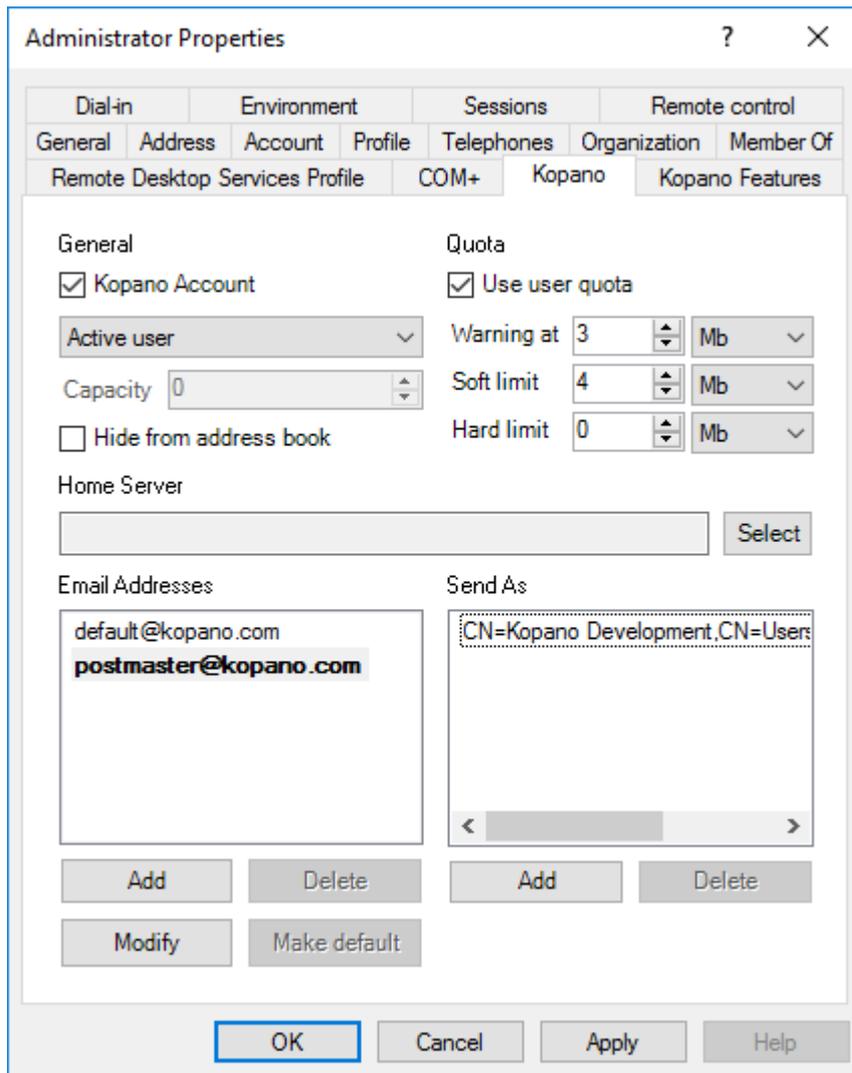


Figure 5.2. Kopano user tab

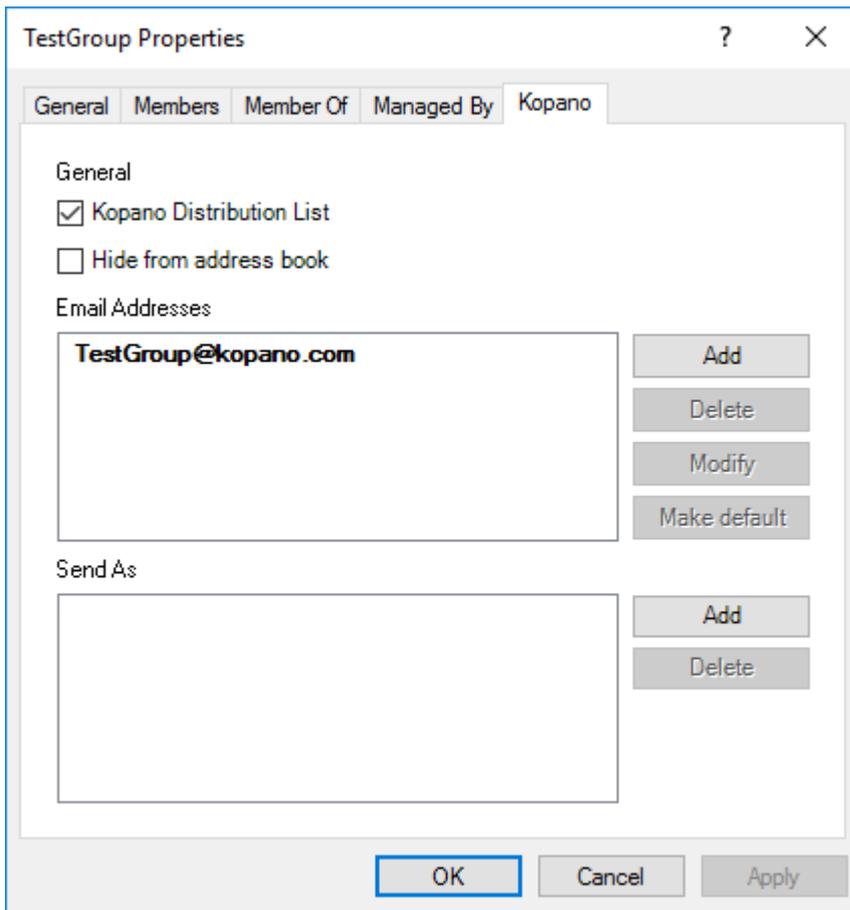


Figure 5.3. Kopano group tab

Note: It is also possible to use the Kopano AD Extension with an existing Zarafa-Schema. The installer still installs the Kopano Schema, to allow an administrator-defined timeframe for moving to the new schema. To enable the Zarafa Schema instead of the Kopano Schema, please modify the registry key HKEY_LOCAL_MACHINE\SOFTWARE\Kopano\Kopano ADS\Prefix from “kopano” to “zarafa”.

Note: Starting with version 1.1 it is possible to store a default email domain in the windows registry. For this set HKEY_LOCAL_MACHINE\Software\Kopano\Kopano ADS\MailDomain to your preferred domain. If the key is missing or empty Kopano ADS will fall back to the default behaviour. See [KC-670](#) for more information.

5.10.4 Configuring Kopano for users stored in LDAP

To manage Kopano users/groups in an LDAP tree the first thing to change is the `user_plugin` setting within `server.cfg`. To enable managing users in an ldap tree, this has to be set to `ldap`.

Editors remark: in earlier versions there was also a backend called `ldapms` for multi server installations, this additional backend has been completely integrated within the `ldap` backend.

```
#####
# USER PLUGIN SETTINGS

# Name of the plugin that handles users
# Required, default = db
# Values: ldap, unix, db
user_plugin = ldap
```

```
# configuration file of the user plugin, examples can be found in /usr/share/doc/
↳kopano/example-config
user_plugin_config = /etc/kopano/ldap.cfg
```

Note: Always remember to switch the server .cfg setting `user_safe_mode` to `yes` when modifying ldap settings to prevent accidental deletes of users.

After the user plugin has been changed the ldap configuration template needs to be copied from `/usr/share/doc/kopano/example-config/ldap.cfg` into the Kopano configuration directory. As see above `kopano-server` expects this file by default at `/etc/kopano/ldap.cfg`.

The following modifications need to be done to your copy of `ldap.cfg`:

Select the matching include directive for your ldap tree:

```
# Select implementation.
# If you have any reason to override settings from /usr/share/kopano/*.cfg,
# do so at the end of this (/etc-resident) config file.
#
!include /usr/share/kopano/ldap.openldap.cfg
#!include /usr/share/kopano/ldap.active-directory.cfg
```

Kopano provides two default configuration files matching the two most common setups. Make sure to select the right include for your ldap backend.

Specify address of your ldap server through `ldap_uri`. For example:

```
ldap_uri = ldap://localhost:389
```

With this setting it is also possible to connect to multiple ldap servers (entries need to be separated by a space), once the first entry is not responding any longer, `kopano-server` will then switch over the the next entry in the list.

```
ldap_uri = ldaps://ldapserver1:636 ldaps://ldapserver2:636
```

Additionally `kopano-server` has to know how to connect the the LDAP server. Kopano will always only read to the LDAP and not write. Therefore a user account with only read access is sufficient.

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

Note: Please be cautious with adding restrictions via ACLs on the `lda_bind_user` for Kopano. It has been observed that excessive restrictions can greatly slow down read performance.

The LDAP search base (base DN) that the search for the different objects should start at. This should be the ‘root’ of the LDAP directory which contains the users, groups and contacts.

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

Any override to configuration options in the included default configuration files should be added to the end of `/etc/kopano/ldap.cfg`.

5.10.5 Fine-tuning user configuration

The following chapter is only necessary if you want to override the default configuration.

In the default configuration a user store is created for each object in the LDAP directory that has a matching `ldap_user_type_attribute_value` attribute (`posixAccount` in `openLDAP` and `user` in `ActiveDirectory`). 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 = (kopanoAccount=1)
```

All user related fields can be mapped by the following options (example values taken from the `openLDAP` configuration):

```
ldap_emailaddress_attribute = mail
ldap_emailaliases_attribute = kopanoAliases
ldap_fullname_attribute = cn
ldap_isadmin_attribute = kopanoAdmin
ldap_loginname_attribute = uid
ldap_nonactive_attribute = kopanoSharedStoreOnly
ldap_password_attribute = userPassword
ldap_user_unique_attribute = uidNumber
```

The unique user attribute is the mapping between a mailbox in the database and the actual user in LDAP. Make sure this field is never changed as Kopano will perceive that as a user being deleted (and created), and will therefore orphan the user's store.

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 `kopano-dagent` directly, this needs to be resolved by Postfix.

Extra user information, like addresses, phone numbers and company information can be mapped by an extra configuration file:

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

The specified attributes for users will also be used for 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 [https://technet.microsoft.com/en-us/library/cc737526\(WS.10\).aspx](https://technet.microsoft.com/en-us/library/cc737526(WS.10).aspx).

5.10.6 Fine-tuning group configuration

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

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

For the membership relationships between groups and users, each group object has a group member attribute. This can be configured by:

```
ldap_groupmembers_attribute = memberUid
```

In `ActiveDirectory` environments this usually uses the following configuration:

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

The Kopano Server will by default use the unique user attribute as value of the group member attribute. This can be changed by the group member's relation attribute.

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

Groups are by default so called “security groups”. Security groups are available in the Global Address Book when creating a new email and setting permissions. To downgrade a group to a “distribution group” kopanoSecurityGroup must be set to 0. Distribution groups are only available in the Global Address Book when creating a new email but cannot be used for configuring mailbox permissions.

```
ldap_group_security_attribute = kopanoSecurityGroup
ldap_group_security_attribute_type = boolean
```

5.10.7 Addresslist configuration

Addresslists are groups of users that match a custom condition. These addresslists are showed as subfolders of the Global Address Book.

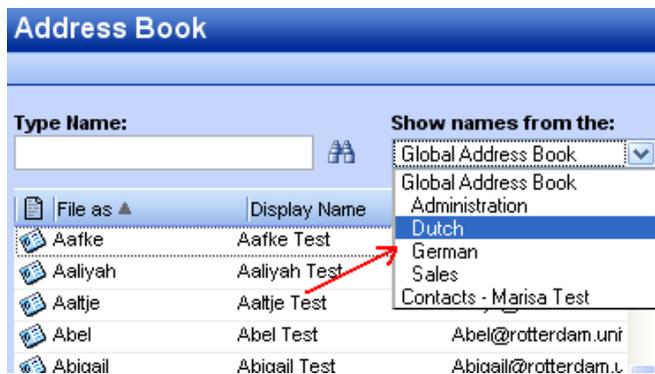


Figure 5.1. Addresslists in Global Address Book

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 = kopanoFilter
ldap_addresslist_name_attribute = cn
```

See the *User Management with LDAP or Active Directory* for more information on how to administer address lists.

5.10.8 Testing LDAP configuration

After the LDAP configuration is done, the changes can be activated by reloading `kopano-server`.

```
systemctl reload kopano-server
```

Important: Only changes to the ldap configuration (mappings, searchbase, bind) can be applied by a reload. For changes to the `user_plugin` in `server.cfg` a proper restart is required.

To test whether users and groups will be listed correctly using the LDAP configuration, use:

```
kopano-admin -l
```

for users, and for groups:

```
kopano-admin -L
```

If no users or groups are shown, please check the Kopano server log file for errors. Setting the `log_level` to `0x00020003` in the `/etc/kopano/server.cfg` will display all LDAP queries send to the server and possible errors.

Note: The first time the `kopano-admin -l` is done, all mailboxes will be created, therefore it can take some time before the execution finishes, so be patient.

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

```
man kopano-ldap.cfg
```

5.11 Postfix integration

Kopano does not include its own MTA, but can easily be integrated with the MTAs found in modern Linux distributions. Although Kopano Groupware is MTA agnostic, we recommend to use Postfix.

In order to deliver an email into a user's mailbox, the `kopano-dagent` is executed. Messages are passed to the `kopano-dagent` from the standard input or by the LMTP protocol. The usage of LMTP is the recommended delivery method as this enables the use of Single Instance Attachment Storage across all attachment backends.

A few examples of the KC 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.

5.11.1 Configure Kopano-dagent for delivery via unix socket

Starting with the 8.7 release of Kopano Groupware Core it is possible to deliver messages not only via a tcp LMTP connection, but also through a linux socket. While for backwards compatibility the default value will continue to use the tcp socket, we recommend to use the unix socket for new installations.

Configuring `kopano-dagent` for delivery via unix socket takes a few steps. Execute the following commands to create a secure local socket, that is only accessible by Postfix:

```
mkdir -p /var/spool/kopano
chown kopano:kopano /var/spool/kopano
chmod go= /var/spool/kopano
setfacl -m u:postfix:rwx /var/spool/kopano
```

After this is setup update/set the value of `lmtplisten` in `dagent.cfg` to make use of the new socket:

```
lmtplisten = unix:/var/spool/kopano/dagent.sock
```

Withing the Postfix configuration (`main.cf`) a matching transport line looks like:

```
virtual_transport = lmtpl:unix:/var/spool/kopano/dagent.sock
```

5.11.2 Configure Postfix integration with OpenLDAP

The Postfix MTA can connect to an OpenLDAP server to resolve the primary mail addresses as well as 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](#) 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 email domains, 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 its 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, ldap:/etc/postfix/ldap-
↳groups.cf, ldap:/etc/postfix/ldap-groups-expand.cf
virtual_transport = lmtp:unix:/var/spool/kopano/dagent.sock
```

All incoming emails are delivered to the LMTP service of the `kopano-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)(kopanoAliases=%s))
result_attribute = mail
```

To deliver mails to member of a group the email addresses of the individual must be resolved:

For resolving group members create the file `/etc/postfix/ldap-groups.cf` and add the following lines:

```
server_host = localhost
search_base = ou=Groups,dc=example,dc=com
version = 3
scope = sub
query_filter = (&(objectclass=kopano-group)(mail=%s))
result_attribute = memberUid
```

To expand group members' mail into uid create the file `/etc/postfix/ldap-groups-expand.cf` and add the following lines:

```
server_host = localhost
search_base = ou=Groups,dc=example,dc=com
version = 3
scope = sub
```

```
query_filter = (&(objectclass=kopano-user)(uid=%s))
result_attribute = mail
```

Note: While this approach creates an additional query, it has the benefit that the `memberOf` overlay does not need to be enabled in OpenLDAP.

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 needs to be restarted:

```
service postfix restart
```

By default the `kopano-dagent` is configured to run as a daemon and started at boot time. With the following commands you can check if the default configuration is used.

For RPM based distributions use:

```
chkconfig kopano-dagent on
service kopano-dagent start
```

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

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

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

5.11.3 Configure KC 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](#) 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 its 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:unix:/var/spool/kopano/dagent.sock
```

All incoming emails are delivered to the LMTP service of the `kopano-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=kopano,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=kopano,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 KC. To use integrate Postfix with distribution groups, Postfix 2.4 or higher is required.

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=kopano,ou=Users,dc=example,dc=local
bind_pw = secret
query_filter = (&(objectclass=group) (mail=%s))
leaf_result_attribute = mail
special_result_attribute = member
```

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 `kopano-dagent` is run as a daemon and started at boot time.

For RPM based distributions use:

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

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

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

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

5.11.4 Configure KC 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 its 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:unix:/var/spool/kopano/dagent.sock
```

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

<code>#Emailaddress or alias</code>	<code>primary mailaddress of user</code>
<code>john@example.com</code>	<code>john@example.com</code>
<code>user1@example.com</code>	<code>user1@example.com</code>
<code>user1@example.net</code>	<code>user1@example.com</code>
<code>alias_user1@example.com</code>	<code>user1@example.com</code>
<code>info@example.com</code>	<code>user2@example.com, user1@example.com</code>

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 `kopano-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 kopano-dagent on
/etc/init.d/kopano-dagent start
```

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

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

Note: It's advised to enable logging of the `kopano-dagent` when running in LMTP mode for monitoring

purposes. To alter logging options for the `kopano-dagent`, adjust the configuration file: `/etc/kopano/dagent.cfg`.

5.11.5 Configure KC Postfix integration with the DB plugin

Alternatively to managing virtual users in a file, the MySQL Database of Kopano can be used to check if a message should be delivered. For this to work most of the configuration for *Configure KC Postfix integration with virtual users* can be reused.

Note: For this to work Postfix needs the ability to do lookups against a MySQL database. In Debian and Ubuntu this can be accomplished by installing the `postfix-mysql` package. When using Red Hat or Centos Postfix doesn't have the `mysql` module included. Alternatively the Postfix Package from the [Centos Plus repository](#) can be used.

Instead of executing `virtual_mailbox_maps` and `virtual_alias_maps` against `/etc/postfix/virtual`, a `mysql` lookup will be defined inside of `main.cf`.

```
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
virtual_alias_maps = mysql:/etc/postfix/mysql-users.cf
```

This lookup is defined as pictured below:

```
# Replace with the user name and password to log into the MySQL server.
user = root
password = kopano
hosts = 127.0.0.1
dbname = kopano
query = select value from objectproperty where objectid=(select objectid from
↳objectproperty where value='%s' limit 1) and propname='loginname';
```

This configuration only resolves the primary mail address of an user. Aliases should be kept in the `/etc/aliases` file or an extra aliases MySQL table.

5.12 Configure Z-Push (ActiveSync for Mobile Devices)

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

Z-Push is an independent project available as an open source from <http://z-push.org/>

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.

Important: Z-Push ≥ 2.1 requires KC 7.0.6 or later.

5.12.1 Compatibility

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. Among others the following devices are known to by working with Z-Push:

- Apple iPhone and iPad
- Windows Phone 7, 7.5 and 8
- Android phones with Android 4.x and newer
- Blackberry PlayBook and 10 (with ActiveSync)
- other ActiveSync compatible devices

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

5.12.2 Security

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 *Configuring SSL for Windows Mobile and Windows Phone* for details).

5.12.3 Installation

Download the latest Z-Push software from <http://z-push.org/download/>

To install Z-Push, simply extract the Z-Push archive to the /usr/share/z-push directory:

```
mkdir -p /usr/share/z-push
tar zxvf z-push-*.tar.gz -C /usr/share/z-push/ --strip-components=1
```

The -C option is the destination where the files need to be installed.

Z-Push is using a state directory to store a per-user synchronisation status and a log directory for its default logging. Make sure that the 'state' and 'log' directories exists and are 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:

```
mkdir /var/lib/z-push /var/log/z-push
chown www-data:www-data /var/lib/z-push /var/log/z-push
```

The user and group name of Apache will differ per Linux distribution. The table below shows an overview of the user and group names of the Apache process.

Table 5.2. User and groupnames per distribution

Distribution	Apache username	Groupname
Red Hat Enterprise Linux	apache	apache
SLES	wwwrun	www
Debian and Ubuntu	www-data	www-data

On systems with SELinux enabled the security context of these folders might need to be changed, e.g.

```
chcon -R -t httpd_sys_rw_content_t /var/lib/z-push
chcon -R -t httpd_sys_rw_content_t /var/log/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 following line to the `httpd.conf` file:

```
Alias /Microsoft-Server-ActiveSync /usr/share/z-push/index.php
```

Make sure that the line is added to the correct part of the Apache configuration, taking care of virtual hosts and other Apache configurations.

Additional PHP Packages

To use the full featureset of Z-Push 2 and the `z-push-top` command line utility, additional php packages are required. These provide SOAP support, access to process control and shared memory.

Table 5.3. Additional packages per distribution

Distribution	Package name
Red Hat Enterprise Linux*	php-cli php-soap php-process
SLES**	php53 php53-soap php53-pcntl php53-sysvshm php53-sysvsem php53-posix
Debian and Ubuntu	php5-cli php-soap

- To install the `php-process` package you need to add an extra channel subscription from the RHEL Server Optional channel.
- The PHP Posix package is included in the SLES SDK Repository.

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

Lastly, make sure that PHP has the following settings:

```
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 you have several php applications on the same system, you could specify the `z-push` directory so these settings are considered only there.

```
<Directory /usr/share/z-push>
  php_flag magic_quotes_gpc off
  php_flag register_globals off
  php_flag magic_quotes_runtime off
  php_flag short_open_tag on
</Directory>
```

If not setup correctly, the smartphone will not be able to login correctly via Z-Push.

Reload Apache to activate these changes.

To use the Z-Push 2.X command line tools, access the installation directory `/usr/share/z-push` and execute:

```
./z-push-top.php
```

and/or

```
./z-push-admin.php
```

To facilitate the access symbolic links can be created, by executing:

```
ln -s /usr/share/z-push/z-push-admin.php /usr/local/sbin/z-push-admin
```

```
ln -s /usr/share/z-push/z-push-top.php /usr/local/sbin/z-push-top
```

With these symlinks in place the cli tools can be accessed from any directory and without the .php file extension.

5.12.4 Mobile Device Management

Users can remote wipe own mobile devices from Kopano WebApp without interaction of the system administrator by using the Kopano MDM plugin.

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

5.12.5 Limiting Access to Certain Users

Starting with Z-Push 2.2.0 it is possible to allow/deny Z-Push access for certain users. For this Z-Push makes use of the enabled/disabled feature functionality of kopano-server (see *Kopano Feature management*). Z-Push uses the keyword “mobile”.

A more in details explanation of this functionality can be found on the Z-Push Wiki <<https://kb.kopano.io/display/ZP/Disable+Z-Push+access>>.

5.12.6 Upgrade

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. 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. States of Z-Push 2.0 and Z-Push 2.1 are not compatible. A state migration script is available in the tools folder.

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.

5.12.7 S/MIME

Z-Push supports signing and en-/decrypting of emails on mobile devices since the version 2.0.7.

Important: Currently only Android 4.X and higher and iOS 5 and higher devices are known to support encryption/signing of emails.

It might be possible that PHP functions require CA information in order to validate certs. Therefore the CAINFO parameter in the config.php must be configured properly.

The major part of S/MIME deployment is the PKI setup. It includes the public-private key/certificate obtaining, their management in directory service and roll-out to the mobile devices. Individual certificates can either be obtained from a local (company intern) or a public CA. There are various public CAs offering certificates: commercial ones e.g. Symantec or Comodo or community-driven e.g. CAcert.org.

Both most popular directory services Microsoft Active Directory (MS AD) and free open source solution OpenLDAP allow to save certificates. Private keys/certificates reside in user's directory or on a smartcard. Public certificates are saved in directory. MS AD and OpenLDAP both use userCertificate attribute to save it.

In Active Directory, the public key for contacts from GAB is saved in PR_EMS_AB_TAGGED_X509_CERT (0x8C6A1102) property, and if you save a key in a contact, it is PR_USER_X509_CERTIFICATE (0x3A701102).

In LDAP public key for contacts from GAB is saved in userCertificate property. It should be mapped to 0x3A220102 in ldap.propmap.cfg (0x3A220102 = userCertificate). Make sure it looks like this in LDAP:

```
userCertificate;binary::
  MIIFGjCCBAKgAwIBAgIQbRnqpxlPa...
```

Important: It is strongly recommended to use MS AD or LDAP to manage certificates. Other user plugin options like db or unix might not work correctly and are not supported.

5.12.8 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.

5.12.9 Troubleshooting

General configuration

Most of the difficulties are caused by incorrect Apache settings. The Apache setup can be tested using a web-browser 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

Please refer to the [Z-Push Wiki](#) on information how to debug synchronization problems.

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!).

In most cases Z-Push Provisioning will work without any issues.

The following message is shown when the provisioning is requested by the server to mobile.

```
POST cmd FolderSync denied: Retry after sending a PROVISION command
```

It’s normal to see this message when a device is reconfigured or e.g. a policy changed. If you see this messages repeated several times (more than 3), then your device is not “understanding” it should execute the provisioning. Follow this procedure to check how to proceed. Look into the Z-Push Mobile Compatibility List to check if this mobile supports provisioning. Some devices like older native Android clients need the LOOSE_PROVISIONING configuration parameter set (see how to set this below).

If your device supports provisioning, try to fully reconfigure the profile on your phone. Try a hard-reset if the error persists

Optionally you may also disable provisioning, however this will also disable the ability to wipe your phone remotely.

Location of the config.php depends on where you installed Z-Push.

Change

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

to

```
define('PROVISIONING', false);
```

Also in order to enable LOOSE_PROVISIONING, change the following:

```
define('LOOSE_PROVISIONING', false);
```

to

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

- **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).

- **Repeated incorrect password messages**

If a password contains characters which are encoded differently in ISO-8859-1 and Windows-1252 encodings (e.g. “§”) the login might fail with Z-Push but it works fine with the WebApp. The solution is to add `setlocale(LC_CTYPE, “en_US.UTF-8”);` to the `config.php` file.

Important: The solution above is for KC 7 and later versions only. KC 6 and earlier versions might not work properly because they lack unicode support.

Special KC Configurations

This chapter describes how to configure special setups that go beyond most common installations of KC.

6.1 Running KC components beyond localhost

When using the SSL connection with certificates it will not only be possible to encrypt the connection, but Linux services will also be able to login using a client SSL certificate.

Repeat the certificate creation script to create certificates for client programs like the `kopano-spooler`, `kopano-monitor`, `kopano-gateway`, `kopano-dagent` and `kopano-admin`. It's possible to create one certificate for all these programs, or a certificate can be created for each program separately. These clients can then login on the SSL connections with their certificate as authentication.

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

Again, when entering the certificate details, at least make the Organizational Unit Name different from the other certificates. Also, do not forget to fill in the Common Name field.

When asked for the creation of the public key, enter `y` and press enter. Now a new certificate called `client.pem` and a public key called `client-public.pem` are present. As an example, the configuration options needed to edit on the `dagent.cfg` file are as follows:

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

Important: For the `kopano-admin` tool to function correctly in a multi-server set-up, a `admin.cfg` file is required in the KC configuration directory, usually `/etc/kopano/`. It also should contain the options mentioned above.

Enter the correct name or IP-address in the `server_socket` option. If Another port number for the SSL connections on the server is used, enter the right port number as well. Replace the password with the password used while creating the certificate.

Copy the `client-public.pem` file to the server location:

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

Now the client knows the private key, and the server knows the public key. The client can login with this key to the server from anywhere on the network or internet.

Note: Be careful with the `client.pem` file. Anybody who has this private key can login to the Kopano server and will be the internal SYSTEM user, who can do anything without restriction.

6.2 Multi-tenancy configurations

This section will provide information regarding the multi-tenancy functionality available with Kopano Core. The feature is available in all editions, but only officially supported in the Enterprise and Hosted editions.

Multi-tenancy mode enables organisations to run multiple organisations on a single KC server where the members of the different organisations won't see each other.

6.2.1 Support user plugins

Multi-tenancy support can only be enabled when using the DB or LDAP plugin. Currently it's not possible to use the Unix plugin. When using the DB plugin, the `kopano-admin` tool can be used to manage tenants (companies), while with the LDAP plugin all information will come directly from LDAP or Active Directory.

Important: The preferred user plugin for multi-tenancy setups is the LDAP plugin.

6.2.2 Configuring the server

The following configuration options in `server.cfg` will be used when enabling the multi-tenancy support.

```
enable_hosted_kopano = false
```

When set to `true`, it is possible to create tenants within the Kopano instance and assign all users and groups to particular tenants. When set to `false`, the normal single-tenancy environment is created.

```
createcompany_script
```

Location of the `createcompany` script which will be executed when a new tenant has been created.

```
deletecompany_script
```

Location of the `deletecompany` script which will be executed when a tenant has been deleted.

```
loginname_format
```

See [Configuring login name](#) for more details about this configuration option.

```
storename_format
```

See [Configuring store name](#) for more details about this configuration option.

Enabling Multi-tenancy

To enable multi-tenancy support in Kopano change the following configuration option in `server.cfg`:

```
enable_hosted_kopano = true
```

Configuring login name

The `loginname` of a user must be unique in order to correctly allow the login attempt. When enabling multi-tenancy support in Kopano, having an unique `loginname` can become difficult as the number of companies (tenants) increases. It is easier when the `loginname` contains the `companyname` as well, to ensure all `loginnames` are unique.

The way the `companyname` is ‘attached’ to the username to create the `loginname` can be configured with the `loginname_format` configuration option in `server.cfg`. This configuration option can contain the following variables:

- `%u` - The `username`
- `%c` - The `companyname` to which the user belongs

As separation character between the `username` and `companyname` a character should be chosen that does not appear inside the `username` or `companyname` itself. Valid characters for example are `@` and `\`.

Some example `loginname_format` for a user named “John Doe” who is member of “Exampleorg”:

- `%u > john`
- `%u%c > john@exampleorg`
- `\\%c\%u > \exampleorg\john`

Although having a `loginname` that contains a `%c` is mandatory for the DB plugin, it is optional for the LDAP plugin. Managing unique `loginname_s` is easier in LDAP because it is possible to use the email address as the `_loginname` attribute. See the LDAP configuration file for more information about the `loginname` attribute.

Note: When passing a username to the `kopano-admin` tool it should be formatted as configured. For example if the `loginname_format` configuration value includes company name variable (`%c`), the company name should be passed to the `kopano-admin` tool every time a username is needed.

Configuring store name

When relations between multiple tenants (companies) are allowed, it is possible that users share their store with users from other tenants. To easily differentiate stores from different tenants, the store name can be formatted to contain the tenant’s name (`companyname`) to which the user/store belongs.

In `server.cfg` the configuration option `storename_format` is provided for exactly this purpose. In the format different variables are provided which can be used to different kinds of information.

- `%u` - The `username`
- `%f` - The `fullname` of the user
- `%c` - The `companyname`, name of the tenant, to which the user belongs

Some examples for a user named ‘John Doe’ who is member of the tenant ‘Exampleorg’:

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

Configuring the LDAP plugin

While when using the DB plugin no additional configuration is required, for the LDAP plugin there are several configuration options that might require changes.

For a multi-tenancy LDAP setup, it is necessary to have the different company in the LDAP tree and below every company container the users, groups and contacts within that specific company. It's not possible to assign a user to a specific company by an LDAP attribute.

See the screenshot below for an example LDAP structure.

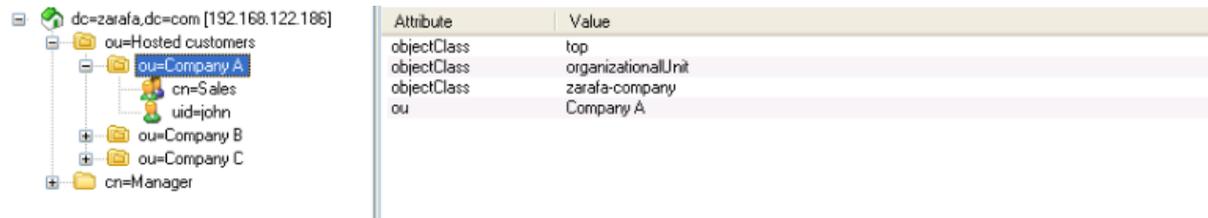


Figure 6.1. LDAP tree multi-tenant environment

Change the following lines in the LDAP configuration file, to configure the multi-tenancy support.

```
ldap_company_unique_attribute = ou
ldap_companyname_attribute = ou
```

Test the settings by using `kopano-admin --list-companies` and `kopano-admin -l`.

If no companies or users are shown, please check the Kopano server log file for errors. Setting the loglevel to 6 in the `/etc/kopano/server.cfg` will display all LDAP queries by the Kopano server and possible errors.

With multi-tenancy support enabled it's not only possible to have different organizations on a single server, but also more advanced settings can be configured, like cross-organization mailbox delegation, different administrator levels and organization quota levels.

See the `kopano-ldap.cfg` man page for more detailed information about these multi-tenancy LDAP features.

```
man kopano-ldap.cfg
```

Public stores

Once the server has been correctly started, stores can be created. There are two type of stores: Private and public stores. There can only be one public store per company space. When creating a company, the public store will be created simultaneously. If for some reason the public store for the specific company is not created, the public store can be created manually by executing the following command:

```
/usr/sbin/kopano-admin -s -I <tenant>
```

Replace `<tenant>` with the name of the tenant (company) for which the public store should be created. When the `-I` option is not used, the public folder will be created for a single-tenancy environment (And will not be accessible when multi-tenancy Kopano is enabled). The public folder is by default available for all users within a tenant (company).

6.2.3 Managing tenant (company) spaces

Note: Management of tenant (company) spaces through `kopano-admin` is only available when using the DB plugin. When the LDAP plugin is used, all administration needs to be done through the LDAP or Active Directory server.

To create a company space use the following command:

```
/usr/sbin/kopano-admin --create-company <companyname>
```

To delete a company space use the following command:

```
/usr/sbin/kopano-admin --delete-company <companyname>
```

To change a company space use the following command:

```
/usr/sbin/kopano-admin --update-company <companyname>
```

This command can be combined with the option `--quota-warn` for setting the quota warning level for the specified company space.

To control the view privileges for company spaces the following commands can be used:

```
/usr/sbin/kopano-admin --add-to-viewlist <viewer> -I <companyname>
/usr/sbin/kopano-admin --add-to-viewlist <viewer> -I <companyname>
/usr/sbin/kopano-admin --list-view -I <companyname>
```

The `<viewer>` is the `companyname` which receives or loses permission to view company `<companyname>`. With the view privileges the Global Address Book can be shared between multiple organizations or use cross organization mailbox delegation.

```
/usr/sbin/kopano-admin --add-to-adminlist <admin> -I <companyname>
/usr/sbin/kopano-admin --del-from-adminlist <admin> -I <companyname>
/usr/sbin/kopano-admin --list-view -I <companyname>
```

The `<admin>` is the loginname of the user who receives or loses admin privileges over the company `<companyname>`. Please note that a user that is administrator over a tenant still needs to be given view privileges to this tenant to see its stores.

6.2.4 Managing users and groups

When using the DB plugin users and groups should be created using the `kopano-admin` tool. For details about using the `kopano-admin` tool see `man kopano-admin`. The user- or group name that should be given to the `kopano-admin` tool depends on the `loginname_format` configuration option.

For example, when `loginname_format` is set to `%u@%c` creating a user for tenant `exampleorg` would be:

```
/usr/sbin/kopano-admin -c john@exampleorg ...other options...
```

And creating a new group for tenant `exampleorg` would be:

```
/usr/sbin/kopano-admin -g group@exampleorg ...other options...
```

6.2.5 Quota levels

When using a multi-tenancy installation there are 2 types of quota, namely the quota for the tenant (company) and the quota for the individual user. The quota for the tenant is checked over the total store size of all users within that tenant plus the public store.

At this time only the warning quota can be configured for a tenant, this means it is not possible to set the soft or hard quota to limit the tenant's email capabilities.

Just like the user quota, there are multiple levels for tenant quota, and there is even a new level for the user quota. A summary of the possible quota levels which can be set in a multi-tenancy environment:

1. Tenant (company) quota:

- (a) Global company quota: Configured in `/etc/kopano/server.cfg` and affects all tenants within the system.
- (b) Specific company quota: The quota level for a tenant configured through the plugin (LDAP or `kopano-admin` tool).

2. User quota:

- (a) Global user quota: This is configured in `/etc/kopano/server.cfg` and affects all users from all tenants.
- (b) Company user quota: This is the default quota level for all users within a tenant, and is configured through the plugin at tenant level.
- (c) Specific user quota: This is the quota level for a specific user, and is configured through the user plugin.

As mentioned above the Global company quota and Global user quota can be configured in the `/etc/kopano/server.cfg` file, in there the options `quota_warn`, `quota_soft` and `quota_hard` for the user quota, and the options `companyquota_warn` for the tenant quota.

To configure the Specific company quota the `kopano-admin` tool can be used when using the DB plugin. The following command will set the various quota levels over the tenant:

```
kopano-admin --update-company <tenant> --qo y --qw <warningquota>
```

To configure the Specific user quota the `kopano-admin` tool can be used when using the DB plugin. The following command will set the various quota levels over the user:

```
kopano-admin -u <user> --qo y --qh <hardquota> --qs <softquota> --qw <warningquota>
```

To configure the Company user quota the `kopano-admin` tool can be used when using the DB plugin by using the `--update-company` argument. The following command will set the various user default quota levels over the tenant:

```
kopano-admin --update-company <tenant> --udqo y --udqh <hardquota> \
--udqs <softquota> --udqw <warningquota>
```

When using the LDAP plugin, the attributes which control the quota levels can be configured in `/etc/kopano/ldap.cfg`.

6.2.6 Administrator users

In a multi-tenancy installation there are two types of administrator users:

- System wide administrator
- Company administrator

The system administrator can access all mailboxes within the hosted environment. A company administrator can only access the mailboxes within the local organisation.

A system administrator can be configured by setting the `kopanoAdmin` attribute to 2 when using LDAP or use `-a 2` when using the DB plugin. A company administrator can be configured by setting the `kopanoAdmin` attribute to 1.

The type of administrator user can be requested by using the `kopano-admin` tool:

```
kopano-admin --details <admin username>
Username:      admin@example.com
Fullname:     Administrator
Emailaddress:  admin@example.com
...
```

6.3 Multi-server setup

This chapter will provide information regarding the multi-server functionality available in Kopano Core.

6.3.1 Introduction

The KC multi-server feature gives the possibility to distribute KC over multiple servers. In this situation the Kopano-user-stores are divided over several servers, but still acting as one central system. The users, groups and tenants (companies) have to be managed in a LDAP or Active Directory server.

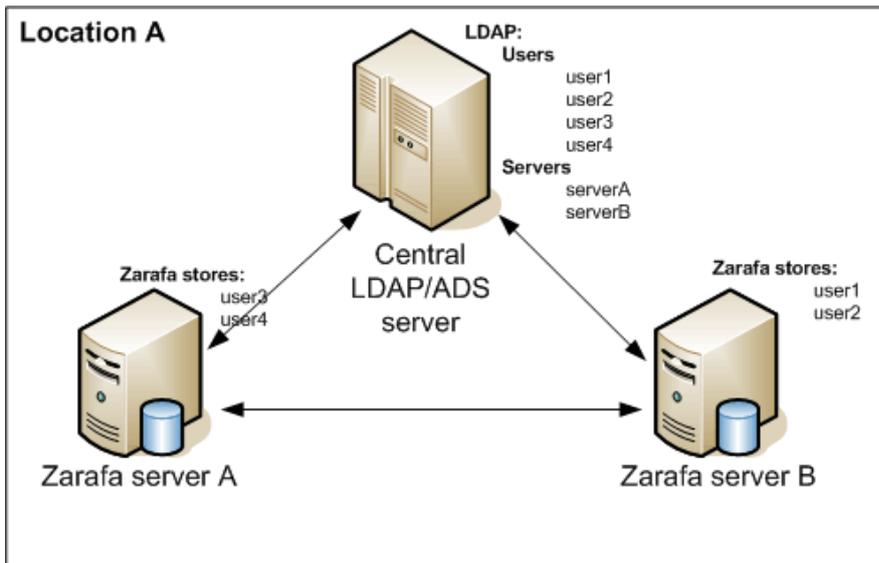


Figure 6.2. Multiserver environment in one location

The multi-server support can also be used to support larger number of users or to spread mailboxes over different geographical locations, see *Multiserver environment on two locations*.

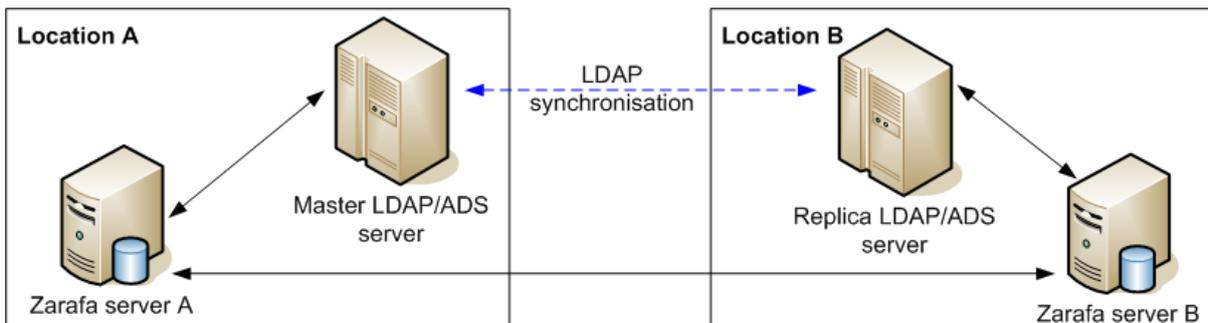


Figure 6.3. Multiserver environment on two locations

The mailbox of a user is always stored on only one server. It's not possible to synchronize mailboxes over multiple servers.

When accessing multiple mailboxes, that are located on different servers, the client will make a connection to the different multi-server nodes. See the flowchart *Multiserver environment*.

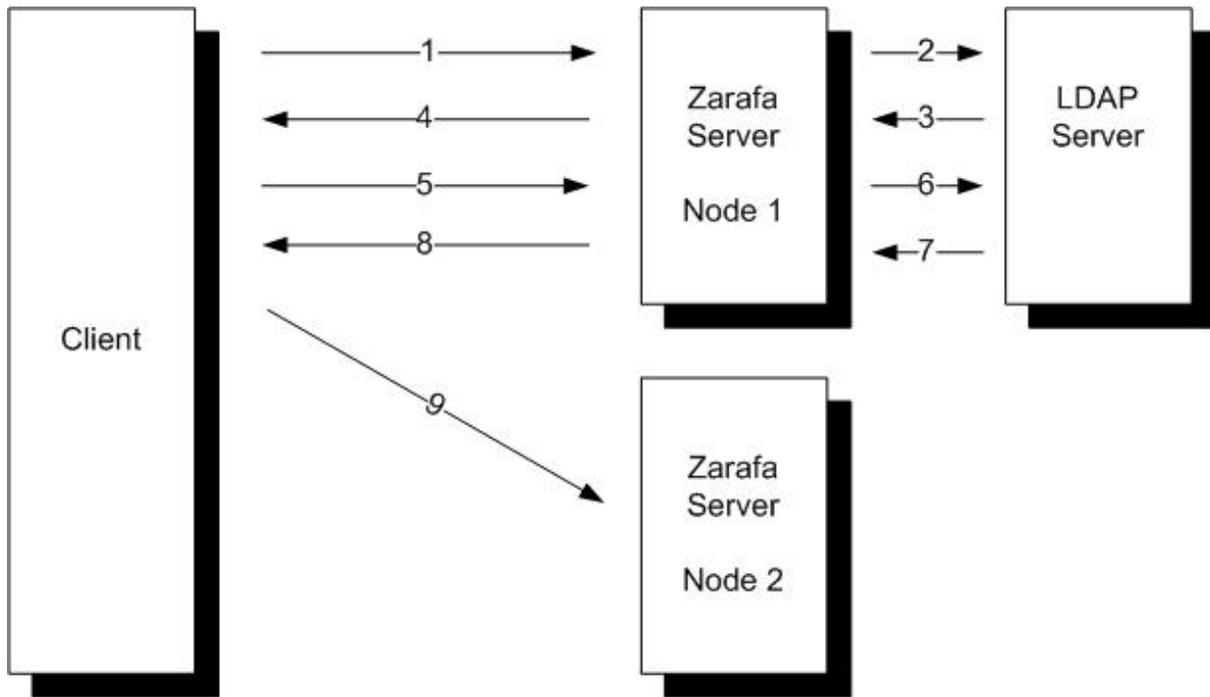


Figure 6.4. Multiserver environment

User *John* is located on *Node 1* and the user *Mary* is located on *Node 2*. John has read access on the mailbox of *Mary*.

1. *John* starts his Outlook client, which connects to *Node 1*.
2. The Kopano Server *Node 1* checks the Home Server attribute in the central LDAP server.
3. The Home Server of user *John* is returned to the Kopano Server.
4. *John's* mailbox is located on *Node 1*, so the mailbox is loaded.
5. *John* sends a request to the Kopano Server to open the mailbox of *Mary*.
6. The Kopano Server *Node 1* checks the Home Server attribute of *Mary* in the central LDAP server.
7. The Home Server of user *Mary* is returned to the Kopano Server
8. A redirect request is send back to the client
9. The client makes a connection to *Node 2* to open the mailbox of *Mary*.

In the above example the client has a connection open to both nodes to access the mailboxes.

6.3.2 Prepare / setup the LDAP server for multi-server setup

The Kopano multi-server version can only be used with the LDAP user plugin.

In a multi-server setup the Kopano Server will not only request user and group information from the LDAP server, but also information about the different multi-server nodes.

1. Setup the LDAP server using *Configure Kopano for user management with LDAP (e.g. OpenLDAP/ADS)* or *Configuring ADS to use the Kopano schema* in this manual.
2. In the LDAP structure add a folder or organizational unit for each Kopano Server node in the multi-server setup.

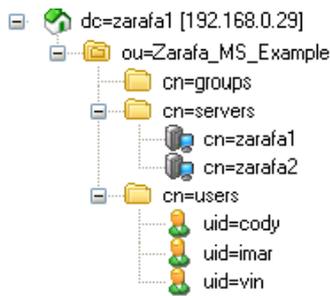


Figure 6.5. Setup directory with all the multi-server nodes

3. Add all the multi-server nodes to this directory or organizational 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 kopano-server objectClass.

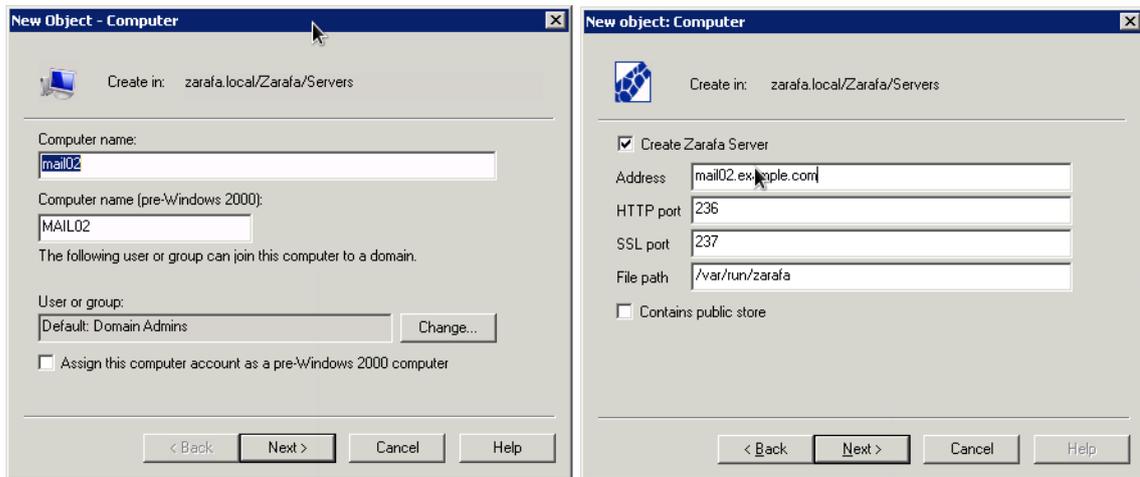


Figure 6.6. Computer creation wizard in ADS

4. Every multi-server node should have a common name, FQDN (recommended) or ip-address and the Kopano server details. Make sure the FQDN can always be resolved by the clients.

Name	Value
cn	ZdsMaster
objectClass	device
objectClass	ipHost
objectClass	zarafa-server
objectClass	top
zarafaContainsPublic	1
zarafaFilePath	/var/run/zarafa
zarafaHttpPort	236
zarafaSslPort	237
ipHostNumber	192.168.0.63

Figure 6.7. LDAP server attributes

5. The attribute KopanoContainsPublic can only be set for one multi-server node at a time. At the moment there is no support for having a single Public Folder onto multiple nodes.
6. The Kopano 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/kopano/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 = kopano-server
ldap_user_server_attribute = kopanoUserServer
ldap_server_address_attribute = ipHostNumber
```

```
ldap_server_http_port_attribute = kopanoHttpPort
ldap_server_ssl_port_attribute = kopanoSslPort
ldap_server_file_path_attribute = kopanoFilePath
ldap_server_search_filter =
ldap_server_unique_attribute = cn
```

1. Every created Kopano user in the LDAP server needs to be assigned to a Kopano server node. This can be set by using the `KopanoUserServer` attribute. The attribute should contain the unique server name.

In a multi-tenancy situation, all created tenants (companies) in LDAP have to be updated with the `kopanoCompanyServer` attribute. Use the server name as well for this.

6.3.3 Configuring the servers

The following configuration options in `server.cfg` are provided for Multi-server support.

```
enable_distributed_kopano
```

Enable multi-server environment. When set to `true` it is possible to spread users and companies over multiple servers. When set to `false`, the single-server environment is created.

```
server_name
```

The unique server name used to identify each node in the setup. This server name should be configured correctly in the DNS. This server name should be the same as the value of the `kopanoUserServer` attribute.

To enable multi-server support in Kopano change the following configuration options in `server.cfg`:

```
user_plugin = ldapms
enable_distributed_kopano = yes
server_name = <servername>
server_listen_tls = *:237
```

Note: An upgrade from single server to multi-server support is not a simple task. Please check with the Kopano Support if migration is possible for the setup used.

6.3.4 Creating SSL certificates

In a multi-server setup, it is required to configure SSL support, because clients like the `kopano-dagent`, `kopano-admin`, `kopano-monitor` need an SSL certificate to login to the different multi-server nodes.

It's required to first create server side certificates, so the Kopano Server is able to accept SSL connections. For the SSL authentication of the Linux clients, like the `kopano-dagent`, a private and public key need to be created.

Follow the steps below to create the required server and client certificates. You need once certificate per server, but could theoretically reuse one client certificates for all client connetions.

1. First, create the directory which will contain the certificates.

```
mkdir /etc/kopano/ssl
chmod 700 /etc/kopano/ssl
```

1. Create the server certificate, by using the `ssl-certificates.sh` script in the `/usr/share/doc/kopano` directory, which uses the `openssl` command and the `CA.pl` script. Before a server certificate can be created a root CA is required. If no root CA is found, the script will first create an own CA.

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

1. Enter a password (passphrase) if you want to use a password for the server key. If a password is set, then this password is needed later on to sign certificate requests. Then enter the certificate information. Give extra attention to the Common Name. This has to be the fqdn of the server and match the value returned by `ldap_server_address_attribute`. The challenge password at the end may be left empty. At the end of the certificate creation the certificate need to be signed against the CA. Accept twice the question for the signing and fill the password of the CA again when asked for.
2. In the last step, the script will ask if it should display the public key of this certificate. This is not necessary, since the certificates have already been created.
3. After completing the `ssl-certificates.sh` script, the server certificate is created in the current directory. The root CA certificate can be found in the same directory or in the default SSL directory of the Linux distribution. On Ubuntu the root CA will be created as `./demoCA/cacert.pem`, on RedHat the root CA will be created as `/etc/CA/cacert.pem`. Edit the following lines in `/etc/kopano/server.cfg`.

Note: The certificate of the server does not necessarily need to be created from the same CA as the client certificate. The CA used for creating the client certificates must be known to kopano-server to be able to validate the certificate chain.

```
server_listen_tls           = *:237
server_ssl_ca_file         = /etc/kopano/ssl/demoCA/cacert.pem
server_ssl_key_file        = /etc/kopano/ssl/server.pem
server_ssl_key_pass        = <ssl-password>
sslkeys_path              = /etc/kopano/sslkeys
```

1. After a restart of the Kopano-server, the server should accept HTTPS connections. Please check the server logfile for any errors.
2. For more options concerning ssl certificates please also see the manpages of `kopano-server.cfg`.
3. If the server certificates are successfully created, the client certificates can be created by the following steps:

```
cd /etc/kopano/ssl
sh /usr/share/doc/kopano/ssl-certificates.sh client
```

1. Fill in all the information, like the server certificate. On some Linux distributions, the Common Name may not be the same as in the server certificate. While the Common Name is important for the certificate creation, it is not of any importance for the Kopano clients. At the end of the creation, it is required to sign again the certificate against the CA and create a public key for the certificate.
2. Two client certificates are created: `client.pem` and `client-public.pem`. The `client.pem` is the private key and will be used by a client (like dagent or spooler). The `client-public.pem` is the public key which is used by the server.
3. Create `/etc/kopano/sslkeys` and move the public key into it.

```
mkdir -p /etc/kopano/sslkeys
mv /etc/kopano/ssl/client-public.pem /etc/kopano/sslkeys
```

1. Restart the `kopano-server` on all nodes to activate the new certificates:

```
systemctl restart kopano-server
```

1. To test the client SSL certificates change the following lines in the `/etc/kopano/dagent.cfg`.

```
server_socket = https://127.0.0.1:237/kopano
sslkey_file = /etc/kopano/ssl/client.pem
sslkey_pass = <ssl-client-password>
```

When the certificates have been set up email can now be delivered by using the ssl socket with the dagent's private-key, in this test case on localhost.

```
kopano-dagent -v -c /etc/kopano/dagent.cfg <username_on_this_node>
Subject: test email
Test
<ctrl-d>
```

Note: Another way of verifying your ssl client configuration is by passing the individual configuration files to `kopano-admin` with the `-c` parameter. This is possible since the ssl options are the same between all client programs. If you are able to query user details of a non-local store your certificate configuration is valid.

When connecting through ssl the dagent will verify the private against the root CA. On Red Hat based systems generated hashed file names have to be created of the root certificates:

```
yum install openssl-perl
cp /etc/CA/cacert.pem /etc/pki/tls/certs/kopano-ca.pem
c_rehash /etc/pki/tls/certs
```

This way the dagent is able to verify the private-key against the CA bundle. On Debian based systems this step can be ignored.

1. If the test case is successful, it is possible to change the following value in the `dagent.cfg` back to:

```
server_socket = file:///var/run/kopano/server.sock
```

1. Deploy the individual certificates to the different multi-server nodes:

```
scp -r /etc/kopano/ssl /etc/kopano/sslkeys root@node2:/etc/kopano/
```

Remember to copy the root CA to the different nodes if this file is placed outside the directories that have just been copied.

1. Repeat the above steps to configure the `server.cfg` and `dagent.cfg` on all the different nodes. On Red Hat based nodes also add the root CA to the CA bundle. When done test remote delivery width:

```
kopano-dagent -v -c /etc/kopano/dagent.cfg <username_on_other_node>
Subject: test email
Test
<ctrl-d>
```

This delivery should not result in any delivery errors, otherwise please check created certificates. It's now possible to deliver email from a central MTA to the different multiserver nodes.

The client SSL certificates can be used for the following tools to connect to a remote Kopano-server:

```
kopano-backup
kopano-admin
kopano-dagent
kopano-search
kopano-spooler
```

For advanced multi-server environments and the best Kopano configuration for a specific setup, the Kopano Professional Services are open for advise and support.

6.4 Single Instance Attachment Storage

The Kopano Server provides Single Instance Attachment Storage to avoid redundant storage of attachments. This feature, as its name implies, only keeps one copy of each attachment when a message is sent to multiple recipients within the same server. This mechanism, thus, minimizes the disk space requirements and remarkably enhances delivery efficiency when messages with attachments sent to large distribution lists.

Let's assume the following situation: user A belongs to a Kopano server; he sends a message with 10 MB of attachments to 30 users that reside on the same server. In a normal situation 30 copies of the files would be saved on the database, leading to an inefficient usage of the storage space (310 MB of data). With single instance attachment store, only one copy of each attachment is saved on the database (only 10 MB of data in this example) and all the 30 users can access the attachment through a reference pointer.

Note: Single instance attachments are accessible between tenants (companies) as well (even when the tenants cannot view each other), the handling of single storage will be transparent. Thus, considering the example above, if user A sends the message to 30 users of tenant1 and 50 users of tenant2, provided that the tenants reside on the same server, only one copy of the attachments is saved.

Note: Single instanced attachments will be handled per server, when sending an email with attachment to multiple Kopano users spread over multiple servers, each server will get its own Single instance attachment.

6.4.1 Single Instance Attachment Storage and LMTP

To use the Single Instance Storage it's required to use the LMTP delivery method executed from the **virtual_transport** in Postfix.

With the aforementioned setup, externally received email with an attachment sent to multiple internal users will be processed efficiently by saving the attachment only once.

The usage of **virtual_transport** in Postfix will deliver only one email with a list of the internal users to the dagent instead of one email per internal user. Without virtual transport option, Single Instance can not know that the attachment is similar in the email item(s).

6.5 Single Sign On with KC

This chapter will describe how to set up a Single Sign On environment with KC, so users can authenticate without entering their password. KC supports both the NTLM and Kerberos authentication protocol.

Both methods will be described in the following sections.

6.5.1 NTLM SSO with ADS

Installing Linux software

The following software needs to be installed:

- winbind
- kinit

Depending on the linux distribution used, this comes through various package names. On Debian use:

```
apt-get install krb5-user winbind
```

krb5-user will also install the Kerberos library configuration files in /etc. The package winbind depends on samba-common which will therefore be installed as well. On Red Hat Enterprise Linux both the krb5-workstation and the samba-common package are required for this.

To enable NTLM SSO with KC set the following option in /etc/kopano/server.cfg:

```
enable_sso = yes
```

ADS: Specific network setup

The following prerequisites have to be met before proceeding:

- Every server must have a DNS name, so their IP-addresses can be found by DNS.
- The time of all servers must be in sync. Time cannot lag for a few minutes.

This document has the following names as example:

- FQDN of the Windows ADS server: `ADSERVER.ADSDOMAIN.EXAMPLE`. Therefore, the windows server is named: `ADSERVER`, the realm is `ADSDOMAIN.EXAMPLE`, and the domain name is `ADSDOMAIN`. Workstations can therefore either join the domain using the `ADSDOMAIN` or `ADSDOMAIN.EXAMPLE` name.
- FQDN of the Linux server is `LINUXSERVER.EXAMPLE`. This name does not matter much, as long as it is handled by the DNS server.

Configuring the Kerberos library

First we are going to configure the Kerberos library. The configuration file is `/etc/krb5.conf`. Under the `libdefaults` section, set:

```
default_realm = ADSDOMAIN.EXAMPLE
```

Under the `realms` section, add the windows realm:

```
[realms]
ADSDOMAIN.EXAMPLE = {
    kdc = 192.168.0.100
    admin_server = 192.168.0.100
    password_server = 192.168.0.100
    default_domain = ADSDOMAIN.EXAMPLE
}
```

Here, `192.168.0.100` is the IP-address of the Windows ADS domain server.

Now that the Kerberos library is configured, it is possible to login using `kinit` on the linux server:

```
kinit Administrator
```

This will ask for a password:

```
Password of Administrator@ADSDOMAIN.EXAMPLE:
```

Type the administrator password there, and a Kerberos ticket should be provided by the ADS server.

Joining the ADS domain

First we'll configure samba. Edit the `/etc/samba/smb.conf` file, and add/set the following options:

For Samba < 3.4

```
[global]
realm = ADSDOMAIN.EXAMPLE
use kerberos keytab = true
security = ads
```

For Samba >= 3.4

```
[global]
realm = ADSDOMAIN.EXAMPLE
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.

With this ticket we can join the Windows domain, without typing the password again:

```
net ads join
```

or if this doesn't work:

```
net ads join -S ADSDOMAIN -U Administrator
```

This command may also be different for different versions of Samba. If this command asks for a password, something goes wrong and it should be killed with `Ctrl-C`. When all goes well, the following line is printed to the screen:

```
Joined 'LINUXSERVER' to realm 'ADSDOMAIN.EXAMPLE'
```

or some other success message.

Now it's required to restart the `winbind` daemon, because it keeps too many items cached:

```
/etc/init.d/winbind restart
```

And that's it. To test if authentication actually worked, try the following command:

```
ntlm_auth --username=john
```

Where `john` is a user on the ADS server.

The program will ask for a password. After entering the password, it should say:

```
NT_STATUS_OK: Success (0x0)
```

If this step does not work, try restarting `winbind`, check the DNS names, check with `strace` what `ntlm_auth` tries to do, check with `tcpdump` if there is actual traffic on the network from `ntlm_auth` to the domain server and other lowlevel debugging tools.

6.5.2 NTLM SSO with Samba 3

Installing Linux software

Depending on the Linux distribution used, `winbind` comes through various package names. On Debian use:

```
apt-get install winbind
```

On Red Hat Enterprise Linux the `samba-common` package is required for this.

To enable NTLM SSO with KC set the following in the `/etc/kopano/server.cfg` file:

```
enable_sso = yes
```

Joining the domain

Now the server need to join the Samba domain by executing the following command:

```
net rpc join
```

Finish by typing the Administrator password. If successful the prompt should give:

```
Joined domain <DOMAIN>
```

The SSO configuration is now done. To test if authentication actually worked, try the following command:

```
ntlm_auth --username=john
```

Where `john` is a valid Samba user.

The program will ask for a password. After entering the password, it should say:

```
NT_STATUS_OK: Success (0x0)
```

If this step does not work, try restarting `winbind`, check the DNS names, check with `strace` what `ntlm_auth` tries to do, check with `tcpdump` if there is actual traffic on the network from `ntlm_auth` to the domain server and other lowlevel debugging tools.

6.5.3 SSO with Kerberos

Requirements and Conventions

- The server that runs KC must have the MIT Kerberos software installed.
- Every server must have a DNS name, so their IP-addresses can be found by DNS. It is also required that all servers have a PTR record.
- The time of all servers must always be in sync with each other.

This document has the following names as example:

- FQDN of the Windows Active Directory Server: `ADSERVER.ADSDOMAIN.EXAMPLE`. Therefore the windows server is named: `ADSERVER`, the realm is `ADSDOMAIN.EXAMPLE`, and the workgroup name is `ADSDOMAIN`.
- FQDN of the Kopano Server is `KOPANO.LINUXDOMAIN.EXAMPLE`.

In this example the Kopano Server is placed in a different domain. This is no requirement, but this makes the document a bit more clear on how to create the Kerberos principal.

Active Directory configuration

Create a Kerberos principal in Active Directory:

1. Add a new user `httpd-linux` to the Active Directory (this user will be used to create the principal for SSO with use for WebApp and DeskApp, username may differ).
2. Make sure that the option *Password never expires* is enabled.
3. On the account properties for these users, enable: *Use DES encryption types for this account*.
4. After setting this account property it is strongly advised to reset the password for these users.

Note: The following commands will use the `kpass.exe` utility, which should be installed by default when the ActiveDirectory is running on the same machine. In any other case you can find it with the “Windows Support tools” on the install cd or download them from the Microsoft website.

Note: When creating a keytab on Windows Server 2008 be sure to specify `RC4-HMAC-NT` as the crypto, `-mapop set +desonly` must be left out.

Execute the following command to create the keytab file for the Apache webserver:

```
ktpass.exe -princ HTTP/fqdn@REALM -mapuser account@DOMAIN -crypto DES-CBC-MD5 \
  -ptype KRB5_NT_PRINCIPAL -mapop set +desonly -pass <password> \
  -out c:\keytab.apache
```

or for Windows Server 2008:

```
ktpass.exe -princ HTTP/fqdn@REALM -mapuser account@DOMAIN -crypto RC4-HMAC-NT \
  -ptype KRB5_NT_PRINCIPAL -pass <password> -out c:\keytab.apache
```

Execute the following command to create the keytab file for the Kopano Server:

```
ktpass.exe -princ fqdn@REALM -mapuser account@DOMAIN -crypto DES-CBC-MD5 \
  -ptype KRB5_NT_PRINCIPAL -mapop set +desonly -pass <password> \
  -out c:\keytab.kopano
```

or for Windows Server 2008:

```
ktpass.exe -princ fqdn@REALM -mapuser account@DOMAIN -crypto RC4-HMAC-NT \
  -ptype KRB5_NT_PRINCIPAL -pass <password> -out c:\keytab.kopano
```

- Copy the file `keytab.apache` to `/etc/apache2` (Debian and Ubuntu) or `/etc/httpd/` (RHEL & SLES) on the Linux server.
- Copy the file `keytab.kopano` to `/etc/kopano/` on the Linux server.

Kerberos configuration

Open the file `/etc/krb5.conf` and insert the following lines:

```
[libdefaults]
  default_realm = ADSDOMAIN.EXAMPLE
  default_tgs_etypes = des-cbc-md5 arcfour-hmac-md5
  default_tkt_etypes = des-cbc-md5 arcfour-hmac-md5
  permitted_etypes = des-cbc-md5 arcfour-hmac-md5

[realms]
  ADSDOMAIN.EXAMPLE = {
    kdc = adserver.adsdomain.example
    admin_server = adserver.adsdomain.example
  }

[domain_realm]
  .adsdomain.example = ADSDOMAIN.EXAMPLE
  adsdomain.example = ADSDOMAIN.EXAMPLE
```

Configuring KC for Kerberos SSO with Outlook:

Add the following line to the `[libdefaults]` section of `/etc/krb5.conf`:

```
default_keytab_name = /etc/kopano/keytab.kopano
```

Kopano Server configuration

To enable Outlook SSO with KC set the following in the `server.cfg` file:

```
enable_sso = yes
```

If the hostname of the Linux server (see the `hostname` command) does not equal the FQDN of the Linux server, the `server_hostname` variable will need to be changed in the `server.cfg` file:

```
server_hostname = kopano.linuxdomain.example
```

Restart the kopano-server to activate all changes.

```
service kopano-server restart
```

Apache configuration (for SSO with WebApp)

Install the mod_auth_kerb/libapache2-mod-auth-kerb Apache module, e.g. for Red Hat:

```
yum install mod_auth_kerb
```

For Debian/Ubuntu: apt-get install libapache2-mod-auth-kerb

Open the vhost configuration of WebApp and add the following lines to the <Directory> directive:

```
<Directory /usr/share/kopano-webapp>
  AuthType Kerberos
  AuthName "Kerberos Login"
  KrbMethodNegotiate On
  KrbServiceName HTTP
  KrbAuthRealms ADSDOMAIN.EXAMPLE
  Krb5KeyTab /etc/httpd/keytab.apache
  require valid-user
</Directory>
```

Set the filesystem permissions of the keytab file to 400 and change the owner to the Apache user:

```
chmod 400 /etc/httpd/keytab.apache
chown apache:apache /etc/httpd/keytab.apache
```

Restart Apache to activate all changes, e.g. for Redhat:

```
service httpd restart
```

WebApp configuration

To setup a Single Sign On environment for Kopano Collaboration Platform, it's necessary to make a trust between the Apache webserver and the Kopano Storage Server. The trust is necessary to manage the user authentication through the webserver and not anymore through Kopano.

There are two ways to establish this trust. The first option is to have the system user running the Apache process acting as an administrator within Kopano, which can only be recommended when Kopano is running on the same system and no other applications (for instance Z-Push) are running on the same server. The second option is to use ssl client certificates (see *Creating SSL certificates*) to establish this trust only for a specific web application.

Using client certificates for authentication

To use ssl client certificates for authentication (see *Creating SSL certificates*) the client certificate has to be readable by the user of the webserver. Afterwards the DEFAULT_SERVER, SSLCERT_FILE and SSLCERT_PASS has to be changed in the config.php of WebApp.

```
// Default Kopano server to connect to.
define("DEFAULT_SERVER", "https://localhost:237/kopano");
```

```
// When using a single-signon system on your webserver, but Kopano is on another
↪server
// you can use https to access the kopano server, and authenticate using an SSL
↪certificate.
define("SSLCERT_FILE", "/usr/share/kopano-webapp/kopano-client.pem");
define("SSLCERT_PASS", mypassword);
```

Running the webserver as an administrator

To have the webserver act as an administrator, the user running the webserver process has to be added on the following line of the `server.cfg`:

```
local_admin_users = root apache
```

Typical users are `apache` for RHEL, `www-data` for Debian/Ubuntu and `wwwrun` for SLES.

Note: This method will only work, when the WebApp is running on the same server as Kopano.

Restart the `kopano-server` processes to activate this change, e.g. for Red Hat:

```
service kopano-server restart
```

Warning: Setting the webserver als `local_admin_user` will allow other applications running on the same server to log in with admin privileges as well. As passwords will no be checked for admin users this means, that user will be able to log in with any password!

Common steps

As the passed user in Single Sign On environments also contains the domain/realm (e.g. `user@domain`), the WebApp has to remove this before logging in. This can be configured in the `config.php` file:

```
define("LOGINNAME_STRIP_DOMAIN", true);
```

Browser configuration

Before Single Sign On can be used in a browser, configure the following settings:

Firefox

1. Type in the addressbar `about:config`
2. Filter on `auth`
3. Change the options: `network.negotiate-auth.trusted-uris` and `network.negotiate-auth.delegation-uris` to `.testdomain.com`

Internet Explorer

1. Go to *Tools > Internet options > Advanced*
2. Make sure the option *Enable integrated Windows authentication* is enabled
3. Add the url of the Kopano Server (`http://kopano.linuxdomain.example`) to the *Local Intranet sites*.

Restart the browser and open the WebApp via the FQDN (`http://kopano.linuxdomain.example`). If the configuration is done correctly, the user will be logged in to the WebApp without typing the username and password.

6.5.4 Up and running

Now that SSO seems to work with the Linux server, it will automatically be used by `kopano-server`. Now log on to a Windows workstation on the domain and use either WebApp or DeskApp to authenticate with the backend, but leave the password field empty.

6.6 Tracking messages with Kopano Archiver

This section provides information on how to track all incoming and outgoing messages using Kopano's Archiving technology. This can be useful in more strict e-mail environments where it's important to be able to see what has been sent and received regardless of what the *owner* of the messages has done with them.

6.6.1 Archive on delivery

Archive on delivery is the process of making sure each message that is received will also be placed in each attached archive. If the message can not be archived it will **not** be delivered. Instead it will result in a temporary failure, causing the MTA to retry the deliver the message at a later time.

Archive on delivery is implemented by the `kopano-dagent` process and can be controlled with the `archive_on_delivery` configuration option in the `dagent` configuration file.

For Archive on delivery to work, an archive configuration file needs to be present in the Kopano configuration directory. In this configuration file settings for `sslkey_file` and `sslkey_pass` must be set to values such that Kopano server can contact the archvier server sucessfully.

When a message is archived with the archive on delivery method, it will become a regular archive entry, meaning the normal rules apply. This means that if a user moves the message in the primary store, the message will also move in the archive. This includes moving to the trash folder.

Important: When a message is deleted from the primary store, the message is **not** deleted from the archive. Instead it is moved to the special Deleted folder in the archiver.

Warning: Due to the current implementation of the Dagent messages that are moved by a rule will sadly be skipped during any subsequent archiving.

6.6.2 Archive on send

Archive on send is the process of making sure each message that is being send by the spooler will also be placed in each attached archive. If the message not be archived it will **not** be send. Instead it will return a failure message to the user.

Archive on send is implemented by the `kopano-spooler` process and can be controlled with the `archive_on_send` configuration option in the `spooler` configuration file.

Important: E-mail that is sent directly to an SMTP server (usually when using an IMAP account) will not be archived directly because the `kopano-spooler` is not involved in the send process in this situation.

When a message is archived with the archive on send method, it becomes a detached archive. This means it has no reference to the original message in the primary store. There's also no message in the primary store that will contain a reference to the archived message.

Note: Unless disabled, messages in the sent items folder are archived as any other message. Extra storage is required because those message have also been archived by the spooler.

6.7 Kopano Python plugin framework

The Kopano Spooler and the Kopano Dagent support the Kopano 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.

6.7.1 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.

6.7.2 General Options

The options for the python plugin framework are for every client the same except the file locations, see [Table Python plugin framework options](#)

Table 6.1. Table Python plugin framework options

Option	Default	Description
<code>plugin_enabled</code>	yes	Enable the plugin framework in the specific component
<code>plugin_manager_path</code>	<code>/usr/share/kopano-<componentname>/python</code>	Path to the plugin manager.
<code>plugin_path</code>	<code>/var/lib/kopano/<componentname>/plugins</code>	Path to the activated plugins.

The value `<componentname>` can be *dagent* or *spooler*

6.7.3 How to use

After the installation of the component kopano-dagent or kopano-spooler, it is possible to activate a plugin. The default plugins are installed in the directory `/usr/share/kopano-<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 following command:

```
ln -s /usr/share/kopano-spooler/python/plugins/disclaimer.py \
    /var/lib/kopano/spooler/plugins/disclaimer.py
```

6.7.4 Kopano-DAgent plugins

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/kopano-dagent/python/plugins/movetopublic.py \
/var/lib/kopano/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/kopano-dagent/python/plugins/movetopublic.cfg /etc/kopano/
↔movetopublic.cfg
```

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/kopano-dagent/python/plugins/BMP2PNG.py \
/var/lib/kopano/dagent/plugins/BMP2PNG.py
```

Note: The package `python-imaging` is required to use this plugin.

6.7.5 Kopano-Spooler plugins

Disclaimer

The disclaimer plugin add a disclaimer to every email sent with the Kopano spooler.

The disclaimer plugin supports plain text and HTML emails. RTF emails are not supported. To use the disclaimer plugin, it is necessary to create the directory `/etc/kopano/disclaimers` which must include the disclaimers. The plugin is using the following files for the disclaimer:

Table 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/kopano-spooler/python/plugins/disclaimer.py \
/var/lib/kopano/spooler/plugins/disclaimer.py
```

6.7.6 Troubleshooting

How to troubleshoot issues you might have while installing or using the Python plugin framework in the Kopano dagent and spooler.

Log explanation

The Python plugin framework can log a lot of information, so if there are issues, it is 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:

```
[TS] [id] PYTHONPATH = /usr/share/kopano-dagent/python/Unknown_path
[TS] [id] Python type: (null)
[TS] [id] Python error: No module named mapiplugin
[TS] [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:

```
[TS] [id] * Loading plugins started
[TS] [id] ! Plugins directory '/invalid/path' doesn't exists. Plugins not loaded.
```

Check the path in `plugin_path` by default it refer to the directory `'/var/lib/kopano/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/kopano-dagent/python/
<DATE> [id] Python type: (null)
<DATE> [id] Python error: 'PySwigObject' object has no attribute 'Log'
<DATE> [id] Python trace: /usr/share/kopano-dagent/python/mapiplugin.py(13) __
↳init__
<DATE> [id] Python trace: /usr/share/kopano-dagent/python/pluginmanager.py(16) _
↳loadPlugins
<DATE> [id] Python trace: /usr/share/kopano-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`

Problem - Solution

- No plugins are loaded in the kopano-dagent Does the plugin exist in the directory `plugin_path` by default in `'/var/lib/kopano/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 kopano-spooler Does the plugin exist in the directory `plugin_path` by default in `'/var/lib/kopano/spooler/plugins'`? If not, create a symlink to the plugin to activated or just copy the plugin into the directory.

6.8 Running KC multi-server behind Reverse Proxy

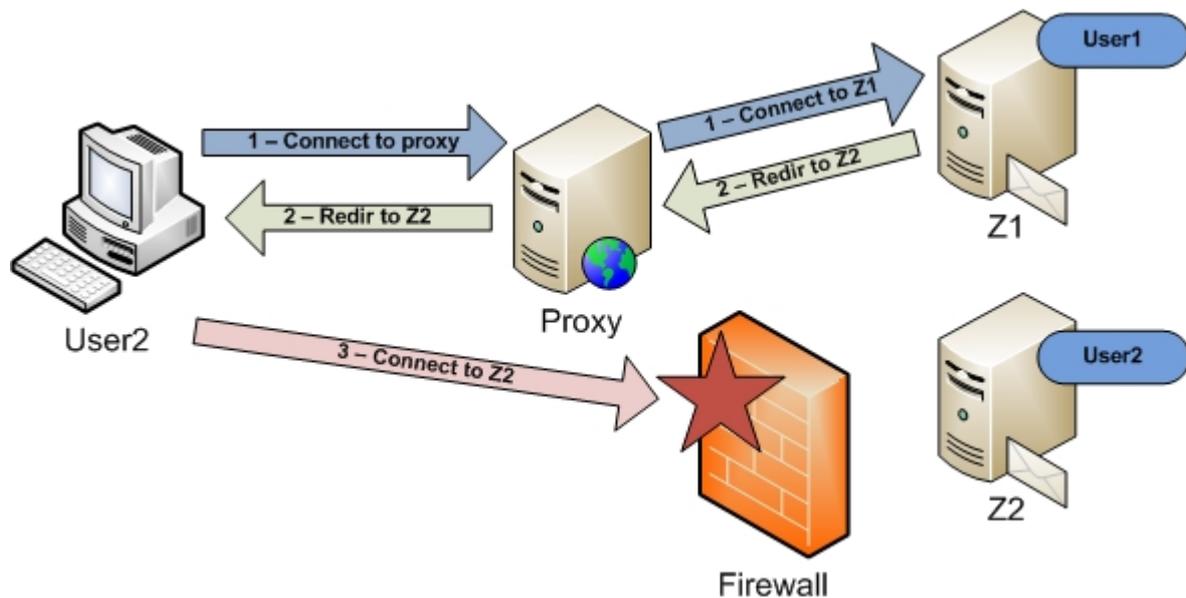
Certain setups require that kopano-server is not exposed directly to the internet. When offering Outlook access, it is sometimes needed to configure a reverse proxy so that Outlook users can connect to the reverse proxy and not directly to kopano-server.

Setting up a reverse proxy with a single kopano-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 Kopano it is quite difficult to setup a reverse proxy for a MutliServer environment, however not impossible.

6.8.1 Description of redirection problem

With redirection the following problem may arise when using a reverse proxy:

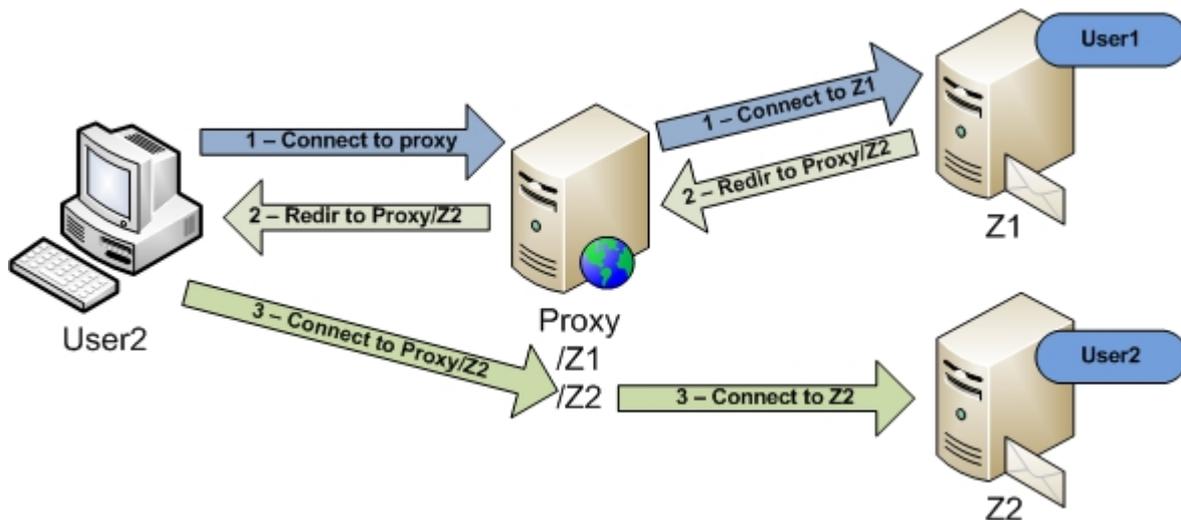
1. Outlook connects to a reverse proxy, and the reverse proxy connects to node Z1.
2. Node Z1 will send a redirect for User2 to node Z2.
3. Outlook tries to connect directly to node Z2, but this connection will break on the Firewall.



In our new setup the reverse proxy will add extra header information, so the kopano-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) Kopano will not reply with the normal redirection string, but it will fetch the connection string from a new ldap attribute: KOPANOPROXYURL.

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.



6.8.2 Setup Prerequisites

When setting up a reverse proxy for a multi-server setup using the new KC options, the following prerequisites need to be met:

1. OpenLDAP or ADS with the schema extensions from KC or newer.
2. A reverse proxy which fully supports HTTP/1.1 (make sure that also the transport encoding “Chunked Encoding” is supported).

6.8.3 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.

Configuring Apache

In our example setup we will create a virtual host which is used for reverse proxying:

1. /kopano will be reverse proxied to node Z1 (Default connection is made to /kopano)
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 /kopano https://z1:237/kopano retry=0
  ProxyPassReverse /kopano https://z1:237/kopano 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 kopano_proxy "yes"
RequestHeader set kopano_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 than the prefork model.

Adding attribute to Servers

We assume you have installed the KC schema extensions.

In ldap add the attribute KOPANOPROXYURL to all servers in the multi-server environment.

For node Z1 this will be:

```
KOPANOPROXYURL: https://zproxy.example.com:237/z1
```

So the complete ldap record for node Z1 may look something like this:

```

objectClass: top
objectClass: kopano-server
objectClass: device
objectClass: ipHost
KOPANOHTTPPORT: 236
KOPANOSSLPORT: 237
KOPANOFILEPATH: /var/run/kopano/server.sock
ipHostNumber: z1.example.lan
cn: z1
KOPANOPROXYURL: https://zproxy.example.com:237/z1

```

For node Z2 this will be:

```
KOPANOPROXYURL: https://zproxy.example.com:237/z2
```

So the complete ldap record for node Z2 may look something like this:

```

objectClass: top
objectClass: kopano-server
objectClass: device
objectClass: ipHost
KOPANOHTTPPORT: 236
KOPANOSSLPORT: 237
KOPANOFILEPATH: /var/run/kopano/server.sock
ipHostNumber: z2.example.lan

```

```
cn: z2
KOPANOPROXYURL: https://zproxy.example.com:237/z2
```

Configuring Kopano Server

Now kopano-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 “kopano_proxy”, if a connection is being made through our reverse proxy.

On all the kopano servers in the multi-server environment we will need to add an extra config option to the server.cfg:

```
proxy_header = kopano_proxy
```

Kopano-server will now send the KOPANOPROXYURL as redirect string to the client when the header “kopano_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 will not connect to the reverse proxy, so the extra header is not added and kopano-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]: kopano-server will check for [header], when found kopano-server send the KOPANOPROXYURL as redirect string.
3. *: will force kopano-server to send the KOPANOPROXYURL 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.

6.9 Running KC with Active Directory in multi-forest environment

To be able to run Kopano Core in a Domain Forest environment, it is recommended to utilize AD’s global catalog. Per default, AD runs it’s global catalog on TCP port 3268, which needs to be configured in your LDAP configuration.

```
ldap_search_base =
ldap_port = 3268
```

Alternatively, using multiple GC ports is also possible by utilizing the LDAP URI configuration mechanism to support multiple GC ports in an AD forest environment, like this:

```
ldap_search_base =
ldap_uri = ldap://ad_with_gc_1:3268 ldap://ad_with_gc_2:3268
```

Please make sure, that the Kopano AD extension has been installed on the schema master of both directory forests. The schema addition automatically installs the new attributes as part of the global catalog. Additionally, this can be verified by checking whether the attributeSchema object has the isMemberOfPartialAttributeSet set to TRUE. More details can be found in the “Global Catalog Partial Attribute Set” section at [https://technet.microsoft.com/en-us/library/how-global-catalog-servers-work\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/how-global-catalog-servers-work(v=ws.10).aspx)

Note: Setting up global catalog also has performance benefits, especially in environments with multiple thousand user objects, since global catalog data is kept in memory. Microsoft Exchange Server for comparison depends by default on the performance of global catalog.

It is possible to resolve groups with members from different, trusted forests. To be able to resolve these, it is required to configure postfix to chase referrals.

```
chase_referrals = yes
```

This needs to be configured with your working ldap-group configuration file or postfix/ldap configuration section (with the above as a suffix).

Since it is required for postfix to have a search_base configured, you are required to define multiple alias map definitions for each forest. Enabling this behavior is configured (with a separated configuration file structure for each LDAP forest) like this:

```
virtual_alias_maps = ldap:/etc/postfix/forest-a-ldap-aliases.cf, ldap:/etc/postfix/  
↳forest-b-ldap-aliases.cf, ...
```

Important: For this functionality it is required to use at least LDAP library version $\geq 2.4.13$ (which is provided by all supported distributions by default).

When Postfix is configured to chase referrals, subsequent binds (for referral query) are done anonymously. This requires anonymous bind to your GC. Enabling this is well described here: <http://support.microsoft.com/kb/320528>

Note: It is recommended to either use `ldapsearch` and/or “`postmap -q ldap:<ldapmap> <test_e-mail>`” to verify the correct resolution of groups.

Using `ldapsearch`, you can verify correct resolution by using first an authenticated search and an anonymous search after:

```
ldapsearch -h <ip-of-an-AD-controller> -p 389 -b "dc=one-of-the,dc=forests" -LLL -  
↳W -x -z0 objectCategory=person -D "cn=Administrator,cn=Users,dc=one-of-the,  
↳dc=forests" dn  
and  
ldapsearch -h <ip-of-an-AD-controller> -p 389 -b "dc=one-of-the,dc=forests" -LLL -  
↳W -x -z0 objectCategory=person dn
```

By comparing the results you can verify the correct result.

6.10 Configuring kopano-spamd for automatic spam/ham learning

kopano-spamd is an optional component to Kopano Groupware Core. It is a local service which listens for changes in a users mailbox and exports mails for spam learning once they are moved into the *Junk E-mail* folder. Once a message is moved out of the *Junk E-mail* folder and back into the *Inbox*, it is exported for ham learning (message wrongly classified as spam).

kopano-spamd by default runs in the user context of the user *kopano* (as defined by *run_as_user* and *run_as_group*). To successfully run *sa-learn* make sure that the provided user is both part of the *amavis* group (so *spamassasin* can process the item), as well as *local_admin_user* within Kopano (so *kopano-spamd* has access to the users mailboxes). The *amavis* group is automatically created when installing Amavis. Installing and configuring Amavis/Spamassasin is not part of this manual. Please change the options *run_as_user*, *run_as_group* and *sa_group* in *spamd.cfg* to match your local environment.

Note: All *kopano-spamd* does is exporting the messages as eml files. These files then still need to be fed to the preferred anti spam engine (for example Spamassasin). This feeding can for example be achieved by watching for changes in the spam/ham folders and kicking off *sa-learn*. See <https://github.com/bkram/inotify-spamlearn> for an example of such a script.

7.1 Starting the services

There are 7 services that can be run:

- `kopano-server`, the main server process
- `kopano-spooler`, sends outgoing email to an SMTP server
- `kopano-monitor`, checks for quota limits
- `kopano-gateway`, provides POP3 and IMAP access
- `kopano-ical`, provides iCal and CALDAV access for clients that use this type of calendar
- `kopano-search`, provides a full text indexing service for quick searching through email and attachments
- `kopano-dagent`, runs as a service when using local mail transfer protocol (LMTP, see *Postfix integration*)

The `kopano-server` and `kopano-spooler` processes are mandatory to run Kopano Core. The `kopano-monitor`, `kopano-gateway`, and `kopano-ical` services are optional. To start a service, type:

```
service kopano-<servicename> start
```

Replace `<servicename>` with the service that needs to start. To start the `kopano-server`, type:

```
service kopano-server start
```

This script will start the server. The `init.d` scripts can start, stop and restart the services. If the `init.d` script cannot be used, the server needs to be started manually. It is possible to explicitly tell the kopano server where the configuration file is, using the `-c` switch:

```
/usr/sbin/kopano-server -c /etc/kopano/server.cfg
```

The `kopano-server` will daemonise, so prompt will almost immediately return. Use `-F` to start it in the foreground. The `-F` switch can also be used for programs like `daemontools` that monitor services.

7.1.1 Stopping the services

To stop a service, type:

```
service kopano-<servicename> stop
```

Most services will stop almost immediately. The `kopano-spooler` may take up to 10 seconds to stop. The `kopano-server` may take up to 60 seconds to stop.

7.1.2 Reloading service configuration

Some options can be modified and reloaded by the service in a live environment. The options that can be reloaded are described in the manual page of the service configuration file. Example: for the `kopano-server`, type the following command to get the configuration manual page:

```
man kopano-server.cfg
```

In the `reloading` chapter are all the options that can be reloaded for that service. To make a service reload the configuration file, type:

```
service kopano-<servicename> reload
```

7.2 Logging options

Each component allows the log method to be chosen in its configuration file. Two ways of logging methods are supported: `file` and `syslog`.

Normally, all KC components log to their respective file located in `/var/log/kopano`. This directory is created when the packages are installed. When this directory is not present, or not writable under the running user, services will not be able to open their log file and will print the log messages to the standard output.

Log messages of the server can be configured. The following options need to be altered in the configuration file:

```
log_method
```

How to log the messages. `file` sends the messages to a file. On Linux systems, `syslog` sends the messages to the default maillog through `syslog`.

```
log_file
```

When the `log_method` is set to `file`, this is the variable that defines the name of file. The server needs write access to the directory and file.

```
log_level
```

Increase the level of messages that will be logged. Level 6 is the highest level.

```
log_timestamp
```

1 or 0; This will enable or disable a timestamp, when using a file as the log method.

Logging of other services than `kopano-server` are configured in a same manner as the server.

7.3 Security logging

Based on this extended security logging auditing can be done on the `kopano-server`. This logging will contain startup messages, user authentications and access actions on delegate stores.

7.3.1 Logging items

Startup

When the server is (re)started, the following message will be printed in the security log:

```
kopano-server startup by user uid=0
```

The following tag is possible in the startup line:

uid The unix user id used to start the server (not necessarily the user the server will be running as)

Signals

When the server receives a signal, the following message will be printed in the security log:

```
kopano-server signalled sig=15
```

The following tag is possible in the signal line:

sig The signal the server received. See `man 7 signal` for a list of most common signal IDs.

Authentications

When a user (not the internal SYSTEM user) logs in, the following message will be printed in the security log:

Correct authentication:

```
authenticate ok user='john' from='127.0.0.1' method='User supplied password' ↵
↵program='apache2'
```

Incorrect authentication:

```
authenticate failed user='john' from='127.0.0.1' program='apache2'
```

Only with sso logins:

```
authenticate spoofed user='john' requested='test' from='192.168.50.178' \
method='kerberos sso' program='OUTLOOK.EXE'
```

The following tags are possible in the authentication line:

user The username sent to the kopano server. requested The name in the MAPI profile to open the store of, user tag will be the actual authenticated user. (SSO only) from Unix socket or IP-address the connection to the server was made to. method Method the user was validated with, one of the following: socket, certificate, password, ntlm sso or kerberos sso. program The program being used to login with.

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** Unix socket or IP-address the connection to the server was made to. **program** The program being used to login with. **impersonator** The user that is impersonating another user. This is the user whose credentials are being checked.

Sharing actions

When a user opens objects that are not within his own store, a message will be logged. This also accounts for the `Public` store.

The following message will be printed in the security log:

Allowed sharing action:

```
access allowed objectid=387538 type=3 ownername='test' username='constant' rights=
↵'view'
```

Denied sharing action:

```
access denied objectid=387538 type=3 ownername='test' username='constant' rights=
↵'view'
```

The following tags are possible in the sharing line:

objectid The object being acted on. **type** The MAPI type of the object. Possible values are 3 (store), 5 (folder) and 7 (message). **ownername** The owner of the store the objectid is in (not necessarily the user that actually created that object) **username** The user performing the action on the object. **rights** The action being performed.

Note: For the `Public` store the ownername will be `SYSTEM` in single-tenancy mode, and the company name in multi-tenancy mode.

Possible actions in rights:

read Reading the object. **create** Creating a new object **edit** Editing an existing object (eg altering properties, but also adding/removing of recipients and attachments) **delete** Deleting (softdelete) or moving the object **create folder** Creating a new folder **view** Reading the folder hierarchy / contents tables **folder permissions** Altering the permissions, modifying and deleting folders **owner** `submitMessage/finishMessage/abortSubmit`, sending email actions in someone else's store is never permitted unless you're the owner. **admin** Unused, will never actually be printed

Log parsing

When a user is accessing a delegate store or folder an entry is written to the `audit.log`. To have a more user friendly overview of the delegate folders accessed, the `audit.log` can be parsed.

The following command will parse the logfile and make the output more user friendly:

```
perl /usr/share/doc/kopano/audit-parse.pl < /var/log/kopano/audit.log
```

The script will display now the exact foldername which is accessed in the delegate store:

```
access allowed rights='view' type='folder' objectid='store\27\IPM_SUBTREE\Calendar
↪' \
        username='john' ownername='mary'
```

In this example the user john has opened the Calendar of user mary.

Not logged

Only “toplevel” objects rights are checked, so you won’t see actions on attachments, recipients or msg-in-msg objects.

7.3.2 Configuration

In the `/etc/kopano/server.cfg` the following options are added:

```
audit_log_enabled = no
audit_log_method = syslog
audit_log_file = -
audit_log_level = 1
audit_log_timestamp = 0
```

By default the audit logging is disabled. When enabled, the default is to log through syslog since this can be configured to send the messages to an external syslog server. The syslog `authpriv` facility will be used to send the messages to.

7.4 Kopano statistics monitoring

The statistics and server status can be checked with the `kopano-stats` tool. The `kopano-stats` tool offers the following options:

- `--system` Gives information about threads, SQL and caches
- `--session` Gives information about sessions and server time spent in SOAP calls
- `--users` Gives information about users, store sizes and quotas
- `--company` Gives information about companies, company sizes and quotas
- `--top` Shows top-like information about sessions and server resource usage

To use the `kopano-stats` tool use for example the following command:

```
kopano-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    kopano-spooler   0:00  0:00    0        6         ↪
↪tableQueryRows
```

The `--top` overview gives every second status information about CPU usage, connected clients, active threads, queue length and SQL queries. When the server has a high queue length and age the amount of threads should be normally increased.

7.5 Soft Delete system

If a user deletes emails, calendar items or complete folders, they are by default moved to the Deleted Items folder.

When the items are removed from the Deleted Items, the items still will not be fully removed from the database. Rather, they are marked as deleted, so the user does not see the items. Even when a user deletes items with <SHIFT> <delete> they are not removed from the database, but marked as deleted.

This makes restoring of items quick and easy from Outlook: choose *Extra* from the menu bar in Outlook menu, and click on *Restore deleted items*. Items are grouped by the folder they were deleted from. Most items will appear in the *Deleted Items* folder as they have been removed from that location.

Soft deletes always remain in the database, until they are purged. When an item will be purged is set by the `softdelete_lifetime` configuration value. The default value is 30 (days).

In this example, the value is set to 30. This means that deleted items will be purged from the database 30 days after they were deleted. When this option is set to 0 (zero), the items will never be automatically removed from the database.

Softdelete purges are automatically run every hour, unless the value for `softdelete_lifetime` is set to 0 (zero). For performance reasons in larger environments, a manual purge of the softdelete system is advisable. This can simply be configured by a cron job.

Purges can be manually triggered with the following command:

```
kopano-admin --purge-softdelete <days>
```

<days> denotes the number of days that recently removed items are kept. When 0 (zero) all removed items are purged.

8.1 Public folder

Once the server has been correctly started, stores can be created. There are two type of stores: Private and public stores. There can only be one public store. It can be created with the following command:

```
/usr/sbin/kopano-admin -s
```

The public store is the folder every user can always open. After installation and configuration of the server a public store needs to be created before private stores can be made. If KC is configured for multi-tenancy, a public store will be automatically created per company.

When using multi-server support, the Public store can only be created on the multi-server node which has the `KopanoContainsPublic` attribute enabled. Currently the Public Store can be created on only one server. See *Prepare / setup the LDAP server for multi-server setup* for more information.

Note: The Public store is by default accessible and writable for all users. Please review the permissions before start using the Kopano system.

8.2 General usage of kopano-admin tool

Kopano offers the `kopano-admin` administration tool for managing user and groups. When using the `DB` plugin the tool can be used to create or delete users and groups. When using the `unix` or `ldap` plugin the tool can't be used for creation of users and groups, but the tool can still be used to get more information about users and groups.

8.2.1 Listing users

All available users or groups can be displayed by using the following commands:

```
kopano-admin -l  
kopano-admin -L
```

8.2.2 Displaying details

To display more information of a specific user, use:

```
kopano-admin --details john
Name: john
Full name: John Doe
Email address: john.doe@kopano.com
Active: yes
Administrator: no
Address Book: visible
Features: mobile; outlook
Store: ABCD1234EFGH5678
Store size: 462.40 MB
Send-as:
Delegation:
Auto-accept meeting requests: no
Out-Of-Office: disabled
Current user store quota settings:
  Quota overrides: no
  Warning level: 1024.00 MB
  Soft level: 2048.00 MB
  Hard level: 3072.00 MB
Groups (2):
  Groupname
  -----
  Everyone
  Sales team

Permissions:
```

To display more information of a specific group, use:

```
kopano-admin --details --type group sales
Name: sales
Email address:
Address book: Visible
Send-as:
Users (2):
  Username      Fullname      Homeserver    Store
  -----
  john          John Doe      Kopano        ABCDE
  mary          Mary Jones    Kopano        FGHIJ
```

8.2.3 Reattaching stores from deleted users

When a user is deleted the mailbox of the user will be still kept in the database. Use the following command to retrieve a list of stores without a user, and users without a store:

```
/usr/sbin/kopano-admin --list-orphans
Stores without users:
  Store guid      Gussed username    Last modified
  ↪Store size
  -----
  ↪-----
  CAC27E6D70BB45B0B712B760AE6BA0A8  steve              2017/05/01 14:22
  ↪2334KB

Users without stores (1):
  Username
```

```
-----  
jane
```

It can be decided to remove the store from the database or hook the store to another user to be able to access it once again. To remove the store from the database, an action which is irreversible, use the following command:

```
/usr/sbin/kopano-admin --remove-store <store-guid>
```

Note: When removing a store, it is not immediately removed from the database, instead it is marked as softdeleted and will be removed as soon as it exceeded the defined `softdelete_lifetime` defined in `server.cfg`. If you want to delete the store permanently, you need to issue the command “kopano-admin --purge-softdelete” as well. Please note that in this case the entire softdelete area will be emptied as well.

To hook the store to another user, use the following command:

```
/usr/sbin/kopano-admin --hook-store <store-guid> -u <user>
```

The user given with the `--user` option will now have the new store attached to it. With the next re-login the new store will be accessed.

Calling `--hook-store` without `--user` will hook the public store.

Important: When a store is hooked to a user that already has a store attached to it, the original store will be orphaned. This original store can be found using the `list-orphans` options of the `kopano-admin` command.

8.2.4 Additional commands and further information

More information about all options of the `kopano-admin` can be found in the man-page.

```
man kopano-admin
```

8.3 Users management with DB plugin

By default the DB plugin will be used as user management plugin. Below will be described how to manage users with the `kopano-admin` command. For user management with the LDAP user plugin, please see [User Management with LDAP or Active Directory](#).

At the moment KC doesn't provide a graphical or webbased user management interface, however there are different 3rd party product that provide webbased management of the Kopano system.

8.3.1 Creating users with DB plugin

To create a new user, use the following command:

```
/usr/sbin/kopano-admin -c <user name> \  
-p <password> \  
-e <email> \  
-f <full name> \  
-a <administrator>
```

The fields between `<>` should be filled in as follows:

- **User name:** The name of the user. With this name the user will log on to the store.

- **Password:** The password in plain text. The password will be stored encrypted in the database.
- **Email:** The email address of the user. Often this is <user name>@<email domain>.
- **Full name:** The full name of the user. Because the full name will contain space characters, and maybe other non-alphanumeric characters, the name should be entered with quotes (' ').
- **Administrator:** This value should be 0 or 1. When a user is administrator, the user will be allowed to open all Kopano stores of any user. It is also possible to pass 2 as administrator level, this will make the user a system administrator who can access mailboxes within other companies.

All fields except the email address are case sensitive.

The password can also be set using the `-P` switch. The password is then not given at the command prompt, but asked for by the `kopano-admin` tool. The password is not echoed on the screen and needs to be typed twice for verification.

8.3.2 Non-active users

A non-active user cannot login to KC, but email can be delivered to this user, and the store can be opened by users with correct permissions. Non-active users can especially used for functional mailboxes, resources and rooms.

To create a non-active user, use the following command:

```
kopano-admin -c -u <user name> -e <email> -f <full name> -n 1
```

8.3.3 Updating user information with DB plugin

The same `kopano-admin` tool can be used to update the stores and user information. Use the following command to update:

```
/usr/sbin/kopano-admin -u <user name> \
                                [-U <new username>] \
                                [-p <new password>] \
                                [-e <email>] \
                                [-f <full name>] \
                                [-a no/yes]
```

All the changes are optional. For example, only the password for an existing user may be updated, leaving the other user information the same as it was.

8.3.4 Deleting users with DB plugin

To delete a user from the server, use the following command:

```
/usr/sbin/kopano-admin -d <user name>
```

The user will be deleted from the database. However the store will be kept in the database, but is not accessible. See *General usage of kopano-admin tool* for more information about handling orphan stores.

8.3.5 Configuring 'Send as' permissions

KC supports two kinds of send delegation:

Send on Behalf permissions

If a user grants the appropriate permission to another user, the latter can send items 'on behalf of' the other user. In this case an email or meeting request will be sent with the following 'from' field: <delegate> on behalf of <user>. This setting can only be set from Kopano WebApp or DeskApp.

Send As permissions

If the system administrator gives the rights to user B to ‘send as’ user A, the receiver of an email will not see that user B sent the email. The receiver will only see the email address of user A in the ‘from’ field.

Setting up sendas delegation with `kopano-admin` is only applicable with the DB or UNIX plugin. For setting up LDAP or Active Directory see *User Management with LDAP or Active Directory*.

Add a user to the list of the delegate being updated as a ‘send as’ user. The delegate can now send mails as the updated users’ name, unless the updated user set the delegate as a user based delegate.

```
kopano-admin --add-sendas <user> -u <delegate>
```

For example:

```
kopano-admin --add-sendas john -u helpdesk
```

Remove a user from the list of the delegate being updated as a ‘send as’ user.

```
kopano-admin --remove-sendas <user> -u <delegate>
```

A list of all user who are delegates is part of the normal user information.

```
kopano-admin --details helpdesk
Name:          helpdesk
...
Send-as:       john
```

Note: With the DB plugin sendas permissions can not be configured on groups.

Note: When both the “send on behalf of” and “sendas” permissions are configured on the same user, the email will always be sent with “on behalf of”.

8.3.6 Groups

The server supports groups. Users can belong to any number of groups. Every user always belongs to the special group Everyone. Defining security settings on folders and items are the same for both users and groups.

For example, the group Everyone has read access to the Inbox of Peter. At this point, every user may read the email in Peter’s Inbox, because all users are a member of the group Everyone.

When a new Kopano user is created, only the free/busy information is open for read access for the group Everyone by default.

Creating groups with the DB plugin

By using the `kopano-admin` tool, groups can be created and users can be added or removed from groups. In the following example, a user john is created, a group administration is created, and the user john is added to the group administration.

```
kopano-admin -c john -p secret -f "John Doe" -e "john.doe@kopano.com"
kopano-admin -g administration
kopano-admin -b john -i administration
```

Using the options `-l` or `-L`, a list of users or groups can be listed from the server.

All created users will be member of the group ‘Everyone’, this can not be changed. Groups created with DB plugin can be used both for configuring permissions and sending emails to a specific group.

8.4 Users management with UNIX plugin

When integrating KC with the default users and groups of the Linux server, some of the user administration has to be done via the default Linux usermanagement tools, like the `useradd` tool and the Kopano specific user administration has to be done with the `kopano-admin` tool.

8.4.1 Creating users with Unix plugin

To create a new user, use the default `adduser` command.

```
useradd <username> -c "Full name"
passwd <username>
```

As the emailaddress of user can't be specified in the `adduser` command, the default email address will be `<username>@default_domain`. The default domain is specified in the `/etc/kopano/unix.cfg`.

This email address can be changed by using the `kopano-admin` tool.

```
kopano-admin -u <username> -e <email address>
```

8.4.2 Non-active users

A non-active user cannot login to KC, but email can be delivered to this user, and the store can be opened by users with correct permissions. Non-active users can especially used for functional mailboxes, resources and rooms.

To create a non-active user with the unix plugin, make sure the login shell of the user is set to `/bin/false`. The login shell for non-active users can be configured as well in the `/etc/kopano/unix.cfg`.

8.4.3 Updating user information with Unix plugin

Changing user information when using the unix plugin can be done for some information with the default Linux user management tools and for other information with the `kopano-admin` tool.

The following information has to be changed in the `/etc/passwd` file or with default Linux user management tools:

- Username
- Password
- Fullname
- Mailbox type (active or non-active)
- Group membership

The following other information has to be changed and configured with the `kopano-admin` tool.

- Email address
- Administrator flag
- Quota
- Sendas permissions

8.4.4 Deleting users with Unix plugin

To delete a user from the server, use the following Linux command:

```
userdel <username>
```

The user will be deleted from the database. However the store will be kept in the database, but is not accessible. See *General usage of kopano-admin tool* for more information about handling orphan stores.

8.4.5 Configuring ‘Send as’ permissions

See “Configuring ‘Send as’ permissions” in the db plugin section.

Note: With the Unix plugin sendas permissions can not be configured on groups.

8.4.6 Groups with Unix plugin

The server supports groups. Users can belong to any number of groups. Every user always belongs to the special group Everyone. Defining security settings on folders and items are the same for both users and groups.

For example, the group Everyone has read access to the Inbox of Peter. At this point, every user may read the email in Peter’s Inbox, because all users are a member of the group Everyone.

When a new Kopano user is created, only the free/busy information is open for read access for the group Everyone by default.

Creating groups with the Unix plugin

Groups can be created and users can be added or removed from groups by the default Linux usermanagement tools. In the following example, the group administration is created and the user john is added to the group administration.

```
groupadd administration
usermod -a -G administration john
```

Using the options `-l` or `-L`, a list of users or groups can be listed from the server.

All created users will be member of the group ‘Everyone’, this can not be changed. Groups created with unix plugin can be used both for configuring permissions and sending emails to a specific group.

8.5 User Management with LDAP or Active Directory

The Kopano-server features a system whereby the administrator of a server can specify an LDAP-based server to retrieve user, group and company information. This means that user management can be simplified for installations and standard LDAP administration tools can be used for user management. Also, using an LDAP server makes it possible to integrate Kopano into an existing environment.

Various LDAP server systems are supported, and Kopano will communicate with any standard LDAP protocol version 3 or later server. This means Kopano works in combination with industry-standard solutions as Microsoft Active Directory, OpenLDAP and eDirectory.

This chapter describes loosely how Kopano uses the LDAP server as a source for user, group, contact and company information. In most cases, the particular setup used will require other options and settings than those described in this document. It is therefore assumed that the reader has a good understanding of how LDAP trees work, and how they are configured in their network.

For more information, please refer to the example configurations and manual pages available on all systems on which Kopano is installed.

Note: Please note that due to performance problems in Samba 4, Samba 4 is not supported as a user source for setups larger than 50 users.

8.5.1 The Kopano user synchronization principle

In any Kopano server, there is a database holding the actual data needed while running Kopano. Apart from the actual folder and item data, the database also holds information on data access rights, user settings, and user meta-data set for users and groups. A lot of this data refers to a specific user ID. For example, an ACL (Access Control List) for the 'inbox' for user A will be stored in the database as a record in the ACL table. This record holds the actual access rights for the objects, and the user ID to whom the access control entry has been assigned.

The user ID stated above is therefore a reference to a user ID within the Kopano database. This ID is stored in the 'users' table, along with a reference to the ID of the user in the external user database (in this case, an LDAP server). For example, user 'A' may have user ID 5 in the Kopano system, and may refer to the item (`dn=cn=user,dc=example,dc=com`) on the LDAP server.

Keeping a list of users in this way also solves the problem of creating the store for a user; There is no way to trigger a store creation event on the Kopano server whenever a user is added in the LDAP server. The 'users' table provides a convenient way to track which users are new to the system and therefore require a new store. The same goes for deleting users, as the user store needs to be removed when the user is deleted.

So, the 'users' table in Kopano is almost exclusively a mapping between the user ID which is used internally in Kopano, and an external reference to a user in the LDAP database. Naturally, when any new users are added or users are removed from the LDAP server, this table must be kept in-sync with the changes.

There are many ways of keeping the 'users' table synchronised with the LDAP server, but Kopano has chosen by default for a 'just-in-time' approach. This means that any time a user is requested from the system, it is first checked in the LDAP server for existence, and then it is checked in the 'users' table for existence. If the user does not exist locally on the Kopano server, then the user is created on-the-fly, before returning the information to the caller.

This means that for users and administrators, the synchronisation seems to be real-time; never will there be a delay between adding or removing users from the LDAP server and the users showing up in Kopano.

Because all Kopano components use the same MAPI interface to connect to the server backend, a situation can't arise with any of the Kopano tools where the user database is out-of-sync. For example, delivering an email to a user that was just created will never fail due to the user not existing in the Kopano users table.

To optimise this synchronisation with very large Global Address Books in LDAP, there is a optional setting `sync_gab_realtime` in the `server.cfg` configuration file. When this option is set to `no` there is no real-time synchronisation between the LDAP directory and the Kopano-server. In this case all Global Address Book entries will be retrieved from the cache of the Kopano-server. This is especially useful for setups which have large addressbooks (more than 10000 entries in the addressbook).

Synchronisation between the LDAP and Kopano server need to be forced with the following command:

```
kopano-admin --sync
```

This command can be executed on daily or hourly basis from a cronjob.

Add/Remove events

The mechanism above creates a situation in which there are six events that can be signaled:

- User creation
- Group creation

- Company creation
- User deletion
- Group deletion
- Company deletion

These six events can be coupled to a script (which will be described later) so that system administrators can perform specific actions on their servers with these events. By default, Kopano will only perform the absolute necessary actions during these events; ie store creation and removal. Any other events can be scripted by the system administrator. This means that by default, no actions are performed during group creation and group deletion.

Group membership

Kopano synchronises users, groups and companies so that it can assign user ID's to them, but the group membership for users is never stored on the Kopano server. This means that group membership changes are real-time also, and the Kopano server will query group membership for a user or a user list for a group directly from the LDAP server. How the mapping between group members and users is done will be discussed later.

LDAP server dependency

Due to the fact that the Kopano 'users' database doesn't actually hold the user or group information, but only a reference to the LDAP server, the Kopano server cannot function without a running and accessible LDAP server. If the LDAP server goes down while Kopano is running, Kopano tools will not be able to perform any actions, as almost all server-side actions require some kind of interaction with the LDAP server. For example, just opening an email requires a query to the LDAP server for the groups that the current user has been assigned to. Only after fetching this information, can Kopano determine whether the current user has the access rights to open the message.

When using OpenLDAP as an LDAP source, it's recommended to use LDAP replication to guarantee that an LDAP server is available at all times by running an OpenLDAP server on the same machine as Kopano. This will make sure that the local LDAP server will always be reachable, and Kopano will always keep running as normal.

Setting up the LDAP repository

While in principle almost any LDAP repository can be used with Kopano, this chapter describes how Kopano requests the data from the server and how that data is used within the Kopano server and tools.

The following information can be read from the LDAP server:

- User details (name, email address, etc)
- Contacts (name, email address)
- Group details (name of group)
- Company details
- User/Group relationships (group membership)
- Company members (users and group membership)
- Company relationships (cross-company view and administrator permissions)

The objects that are classified as users, contacts, groups, dynamic groups, addresslists or companies and the attributes that contain the data can be configured within the Kopano configuration files, so Kopano can meet the LDAP schema needs. However, here are some pointers to keep the LDAP repository clean and easy-to-manage:

- Always use some sort of graphical user interface for user and group management. There are many LDAP configuration tools. (For example, [phpLDAPadmin](#) for OpenLDAP as a web based interface)

- If there are users that will be using Kopano, while other users will not, try to group these users into separate ‘folders’. An OU record or any other `dc-type` object can be used to create these folders.
- If Microsoft Active Directory is run, make sure that the real users are in a separate LDAP folder so that Kopano doesn’t need to import the standard users like ‘Administrator’ and ‘Guest’ into the database. It is also possible to filter the users using an LDAP search query, but these search queries can become unsatisfactorily large when using ADS.

As a general rule, always use the LDAPS (SSL) protocol while contacting the LDAP server. When SSL is not used, information will be transmitted clear-text over the wire. This opens possibilities to sniffing user (and administrator!) passwords from the network wire. Kopano supports connecting through LDAP via SSL and a certificate specified in `/etc/ldap/ldap.conf` which is compatible with both Microsoft Active Directory as OpenLDAP servers.

Note: Please note that if plan to set profile pictures for the users, the `jpegPhoto` and `thumbPhoto` attributes have a limit size. Surpassing these limit sizes can cause issues, especially for offline profiles. These limit sizes for these attributes are shown in the `proppmap.cfg`.

8.5.2 User management from OpenLDAP

Creating users using OpenLDAP

Users and groups can be created by using a standard OpenLDAP administration for example `phpldapadmin` or the Windows tool `ldapadmin`.

To configure Kopano specific information for the user, the objectClass `kopano-user` has to be added to the user. Adding this objectClass enables you to add Kopano attributes to the user, like quota settings, sendas permissions, mailbox type.

Creating groups using OpenLDAP

Created groups in OpenLDAP will be used by default as security groups in KC. The security groups can be used for settings permissions and sending emails. Distributions groups can only be used for sending emails and will not be displayed when setting the security permissions on a folder.

To switch a group to a distribution group the attribute `kopanoSecurityGroup` has to be set to 0.

Creating contacts using OpenLDAP

The Global Address Book can be extended with contacts. Contacts are typically external SMTP addresses and can be used as members of `distributionlist`. Contacts can have the same additional attributes (Telephone, ..) as normal users.

Contacts must have the same unique attribute as users. Please check the `ldap_unique_user_attribute` in the `ldap.cfg` for the correct attribute.

Configuring sendas permissions using OpenLDAP

Sendas permissions can be configured both on users and contacts. The users or groups that should be able to sendas a specific address, need to be added in the `sendas privilege list`.

To check whether the permissions are correctly set, use:

```
kopano-admin --details <username>
```

For example:

```
kopano-admin --details helpdesk
Name:          helpdesk
...
Send-as:      john
```

The users that have the sendas permissions, should now be able to add the other address in the 'FROM' field and 'sendas' this account.

'Sendas' permissions have to be configured on the user which is select as the FROM address. Groups can also be used for setting sendas permissions.

Note: When using groups for the sendas permissions, make sure the `ldap_sendas_attribute_type` is set to `dn`. See the following LDAP configuration:

```
ldap_sendas_attribute = kopanoSendAsPrivilege
ldap_sendas_attribute_type = dn
ldap_sendas_relation_attribute =
```

Setup addresslists in OpenLDAP

Addresslists are subsets of the Global Address Book that match a specific criteria. For example, you can create an address list that contains all users in Manchester and another that contains all users in Stuttgart.

To setup an addresslist in OpenLDAP, follow these steps:

1. Create an Organisation Unit for all the addresslists in the LDAP tree.
2. Create a new LDAP object and add the objectClass `kopano-addresslist`
3. Set the `cn` attribute to the unique name of the addresslist
4. Create a condition query in the `kopanoFilter` attribute, see *LDAP Condition examples* for example condition queries.

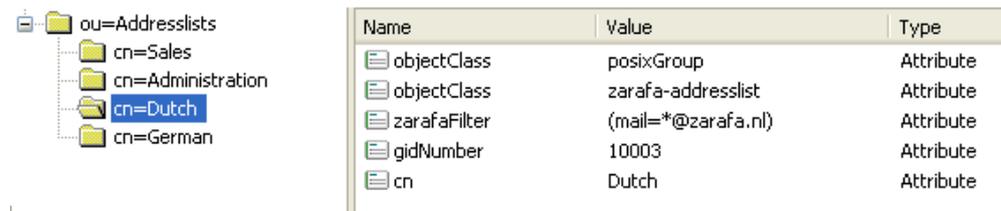


Figure 8.5. Addresslists in LDAP

After restarting the `kopano-server`, the addresslists should be visible in the global addressbook.

Hide information from Global Address Book with OpenLDAP

With KC it is possible to hide users, contacts or groups from the Global Address Book.

Hiding information from the Global Address Book can be done by setting the `kopanoHidden` attribute in OpenLDAP to `1` on a specific object.

Note: The internal System user and the Everyone group can be made hidden in the `/etc/kopano/server.cfg`.

8.6 LDAP Condition examples

For both addresslists and dynamic groups a LDAP filter need to specified. For example, the Global Address Book contains Dutch and German users. It is possible to view these users per country by creating two addresslists in the LDAP tree. All German users have the domain *example.de* in the mail address, and all the Dutch have *example.nl*.

In this situation, the condition `(mail=*@example.de)` is used for the addresslist German, and `(mail=*@example.nl)` for the addresslist Dutch.

Any combination with LDAP attributes are applicable. This following example selects everyone that is a Kopano administrator and has the character `p` in the `cn` value.

```
(&(cn=*p*)(kopanoAdmin=1))
```

This example selects all users with mailaddress `piet@example.de` or `klaas@example.nl`.

```
(|(mail=piet@example.com)(mail=klaas@example.com))
```

8.7 Kopano Feature management

Some features within KC can be disabled. By default, all features are disabled. Enabling can be done globally or on a per-user basis. When a feature has been globally disabled, you may enable the feature in a per-user basis too. Currently the only features that can be controlled are 'imap', 'pop3', 'mobile' and 'outlook'.

If the 'pop3' feature is disabled, users won't be able to login using the POP3 protocol. The same goes for the 'imap' feature, but this has an extra effect aswell. When a user receives email when the 'imap' feature is enabled, the original email and some other imap optimized data will also be saved in the Kopano database and attachment directory. This will make the IMAP services provided by the kopano-gateway more reliable. On the other hand, it will also use more diskpace. Disabling the 'imap' feature will thus save diskpace.

The following table will show when a user can use IMAP or POP3.

Table 8.1. Access control overview

	Service enabled for user	Service disabled for user	Nothing configured for user
Service listed in <code>disable_feature</code> in <code>server.cfg</code>			
Service not listed in <code>disable_feature</code> in <code>server.cfg</code>			

8.7.1 Globally enabling features

To enable a specific feature, edit the `disabled_features` setting in your server configuration:

```
disabled_features = imap pop3 mobile outlook
```

8.7.2 Per-user en- or disabling features

Managing the feature per user depends on the user plugin which is used. For the `db` and `unix` plugin the `kopano-admin` tool has to be used to control the features:

```
kopano-admin -u john --enable-feature imap
kopano-admin -u john --disable-feature pop3
```

For Active Directory or OpenLDAP setups (using the `ldap` or `ldapms` user plugin), the features will be managed from two LDAP attributes `kopanoEnabledFeatures` and `kopanoDisabledFeatures`. Make sure the latest schema file or Active Directory plugin is installed, before using these attributes. These multi-valued attributes can contain any string, but only the features Kopano knows about will actually be provided through the system.

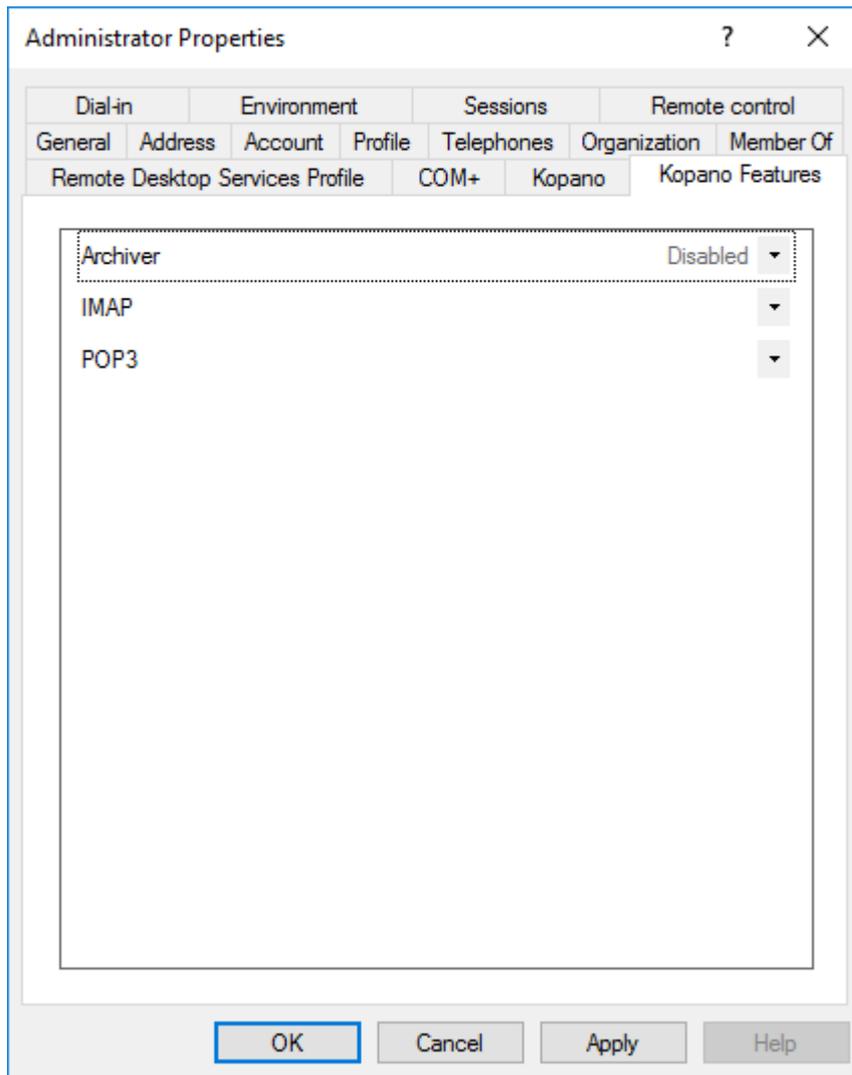


Figure 8.6. Kopano features tab in ADS

Note: Make sure a particular feature isn't listed in both `kopanoEnabledFeatures` and `kopanoDisabledFeatures`. Consistency will not be guaranteed.

8.8 Resource configuration

KC supports automatic booking of resources, like beamers, rooms or other equipment. To create a resource add a new non-active mailbox or select in Active Directory or OpenLDAP the resource user type.

Before a resource can be booked by users, the resource has to be configured to automatically accept meeting requests. The automatic acceptance of meeting request can be configured using the kopano-admin tool.

To configure the resource from Outlook, use the following steps:

- Make the resource temporarily active
- Login as the resource in Outlook
- On the Tools menu, click Options, and then click Calendar Options.
- Under Advanced options, click Resource Scheduling
- Enable the automatic acceptance of meeting request
- If the resource should decline double bookings of the resource or bookings of recurrent meetings, the options “Decline recurring meeting request” and “Decline conflicting meeting requests” should be enabled.
- Configure the permissions on the calendar of the resource, so the users can book the resource. Users should have at least write permissions to the calendar of the resource.

To configure the resource with the kopano-admin tool, use the following command:

```
kopano-admin -u <resource name> --mr-accept yes
```

The resource will now automatically accept meeting requests. To decline double booking or recurrent meeting, use:

```
kopano-admin -u <resource name> --mr-decline-conflict yes
kopano-admin -u <resource name> --mr-decline-recurring yes
```

After the automatic acceptance of meeting requests is configured, make sure the users have at least write permissions on the calendar of the resource. The permissions can be configured by opening the resource mailbox to an administrator user and setting the permissions.

To automatically book a resource make sure the resource option is really selected in the Freebusy times when scheduling the meeting.

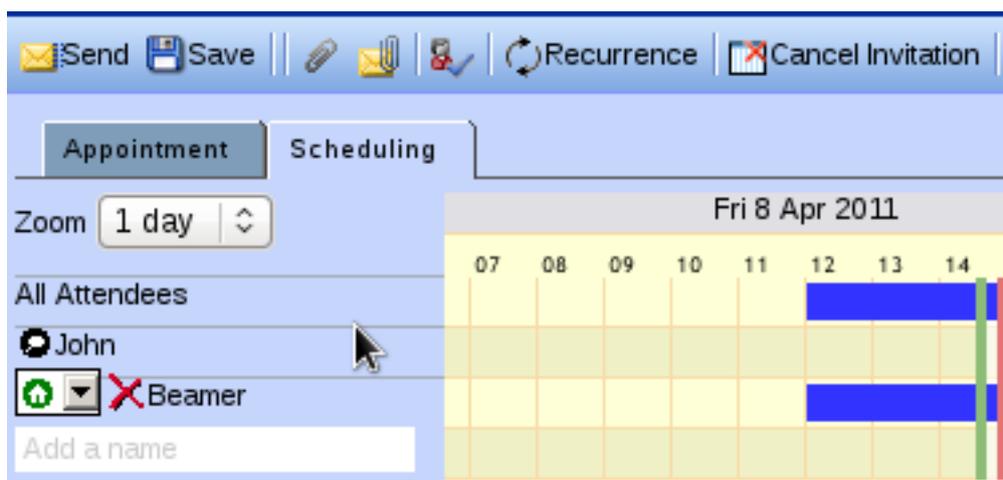


Figure 8.7. Resource option in Freebusy times

8.8.1 Resource booking methods

There are two methods for booking resources:

1. Direct booking
2. Meeting-request booking

Both methods are used to book resources; The final outcome is that the user can book a resource, after which the resource's calendar will show that it is busy for the allocated timeslot. Both methods support declining recurring and conflicting meetings, but the way that they work differ in various ways:

Table 8.2. Table Comparison of resource booking methods

Direct booking	MR booking
Books directly in target calendar	Sends meeting request which is responded to
Needs read/write access to resource's calendar	Needs no read or write access to resource's calendar
Possible to limit bookers through permissions	Not possible to limit bookers
Does not support multiple resources using the same calendar	Possible to set double-booking limit to 2 or higher for equipment
Doesn't work with external bookers	Works with external bookers

8.8.2 Meeting request (MR) booking

Booking by meeting requests works exactly the same as sending a meeting request to another user; When booking the resource, a user sends a meeting request to the resource in an e-mail. The resource then receives the e-mail, checks its own availability and replies to the meeting requests just like a human user would; the booker receives an *Accepted* or *Declined* meeting response by email.

This means that when the meeting is sent to the attendees, the resource has actually not been booked yet; it is possible that another user has booked the resource in the mean time, resulting in a *declined* response from the resource. The booker must then re-schedule and send all participants an update.

The main advantage of this method is that the booker needn't have write permissions on the resource's calendar. Also, the MR method allows for more flexible handling of meeting requests. For example, if the user has 5 projectors, which have been created as a *resource*, then they could be created as 5 separate resources, each of which would normally be directly booked. However, this would require the user to search for a free projector and book that specific projector.

With MR booking, the administrator can set the equipment's capacity to a number other than 1, for example 5 in this case. The administrator then only needs one resource with a capacity of 5 to represent all the projectors. When the MR is processed by the resource, it will check whether *all* projectors were booked at that moment, only declining when all 5 projectors were not available at that moment.

Please note that you *must* use the *equipment* type for your resource if you wish to use the capacity feature. The capacity of *room* resources is ignored (you can not double-book a room).

MR booking is processed by the `kopano-mr-accept` script which is installed by default. This script is triggered by `kopano-dagent` in both direct and LMTP mode when the destination user's `mr-accept` setting is set to TRUE AND the incoming message is a meeting request or meeting cancellation. If the `kopano-mr-accept` script fails, delivery processing is done as usual, possibly triggering delivery rules and out-of-office messages.

Note: In rare cases `kopano-mr-accept` prints out a warning about using `localtime()`. This relates to the - per default - unspecified `date.timezone` variable of `php.ini`. Setting it to for example `date.timezone = Europe/Berlin` fixes these messages.

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 KC ships a commandline utility to manage out of office replies.

To use the utility, use the following command:

```
kopano-set-oof -u <username> -m 1|0 -t "Out of office subject" -n <path to out of_
↳office text>
```

To enable an out of office reply for the user john use:

```
kopano-set-oof -u john -m 1 -t "I'm on holiday till the 30th of June" -n /tmp/oof.
↳txt
```

Other options can be gathered from the help of the script. This can be reached when the script is called without any arguments.

Performance Tuning

When installing a Linux server with Kopano, it is imperative that MySQL is correctly configured to achieve maximum performance on the server; almost all performance bottlenecks are within the database access itself, so getting the SQL queries to run as quickly as possible is very important.

For large installations, it is strongly advised to tune Kopano's cache parameters as well; These are normally set quite low to make sure that Kopano can run on relatively low-end servers, but in anything but the smallest installations, these defaults needs to be upped. Any installation with 50 or more users should definitely tune the cache parameters for maximum performance.

This document assumes the primary role of the server is to run Kopano. Always make sure that other factors are taken into account - for example an anti-spam system or a webserver running a site other than the Kopano WebApp.

There is also a more advanced tuning guide available to customers with a valid subscription. Please create a ticket with the Kopano support to retrieve it.

9.1 Hardware Considerations

There are also various different hardware setups to consider when setting up a server for Kopano. We will discuss the various types of hardware that affect performance.

9.1.1 Memory usage

Tuning memory usage is one of the best ways of increasing server performance; as RAM is generally cheap, using a large amount of RAM in the server properly can boost performance by orders of magnitude.

On the other hand, setting RAM usage too high may cause the server to swap out parts of the memory which need to be swapped back in later, causing a large slowdown in all parts of the server. It is therefore important to set the RAM usage of various components to a high enough setting to use the RAM available, and at the same time not to set the RAM usage too high.

To make use of the available RAM as best as possible, Kopano is designed to use only a fixed amount of physical RAM; the memory usage does increase per user that connects, but only by a small amount - the largest part of the memory usage is due to cache settings in the configuration file. This makes it very easy to control the exact amount of memory that will be used in a live situation, and one can be pretty sure that the actual amount of RAM used will never go far beyond the values set.

So, in general, the optimum RAM usage is as high as possible, without making the system needing to swap out important parts of available memory.

It is very difficult to give a fixed value for what the optimal memory usage distribution is for a given server, as data access patterns vary wildly from server to server. We will describe some rule-of-thumb parameters here and make the RAM usage patterns as clear as possible here.

9.1.2 Hardware considerations

In servers running Kopano, the main performance bottleneck will be the route between the data on the hard disk, and the time it takes to get to the client. This means that generally, I/O performance is more important than CPU performance. Using this as a basis, the following pointers may help in selecting the correct hardware for the system:

9.1.3 More Memory is More Speed

More RAM means better caching and therefore better speed.

Kopano is specifically designed to make use of the large amounts of RAM that is available in modern servers. On the other hand, please remember that in normal Linux server the maximum amount of usable RAM in a 32-bit server is 3Gb unless PAE (physical address extension) is supported in the kernel, CPU and mainboard. If more than 3Gb is needed without some sort of limitation, use a 64 bit system, a 64 bit Linux OS, and a 64 bit Kopano package.

9.1.4 RAID 1/10 is faster than RAID 5

In general, a RAID1 or RAID10 array is faster at database accesses than RAID5 and RAID6. Kopano strongly recommends not use the RAID5 or RAID6 configuration to prevent performance issues.

9.1.5 High rotation speed (RPMs) for better database performance

High-end SCSI or SAS disks regularly have high rotation speeds of 10K or even 15K RPMs. The rotation speed of the disks affects seek times on the disk. Although the Kopano database format is optimized to have data available on the disk in a serial fashion, and most reads are done fairly localized on the disk, seek time is still a large speed factor for I/O. The higher the rotation speed, the lower the seek time.

9.1.6 Hardware RAID

Hardware RAID controllers often have large amounts of cache RAM. This can also increase performance and data throughput of the I/O subsystem. If a hardware RAID controller is used however, always make sure that either write-back cache is not used, or a functioning UPS and shutdown process for the server are available, as write-cached data will be lost when the power fails. This is not only harmful for the data that was written at that moment, the write could actually corrupt the on-disk innodb data.

9.2 Memory Usage setup

There are basically 4 large parts of the server setup that use server memory:

- Kopano's cell cache (caches individual cell data within a table view)
- MySQL's buffer size (caches reads and writes from the ibdata file)
- MySQL's query cache (caches exactly repeated SQL queries)

In a server purely running Kopano, make sure these caches are setup to use around 80% of the RAM in the server. The other 20% should be free for system processes, other processes (like MTA) and the webserver.

For a general rule-of-thumb, the following RAM distribution should be used:

Kopano caches:

- **cache_cell_size**: around 25% of total RAM size
- **cache_object_size**: about 100kb per user
- **cache_indexedobject_size**: about 512kb per user

These cache settings need to be configured in the `/etc/kopano/server.cfg` file. To activate the cache size changes the Kopano Server need to be restarted.

MySQL settings:

- **innodb_buffer_pool_size**: around 50% of total RAM size
- **mysql_query_cache**: 32Mb
- **innodb_log_file_size**: 25% of the `innodb_buffer_pool_size`
- **innodb_log_buffer_size**: 32M
- **innodb_file_per_table**: 1
- **max_allowed_packet**: 16M
- **table_cache**: 1000

These settings need to be configured in the `/etc/my.cnf` or `/etc/mysql/my.cnf` file below the `[mysqld]` section.

It's recommended to change these MySQL settings before starting the Kopano Server and migrating user data.

The most important settings will now shortly be described to illustrate the need of each of these cache settings.

9.2.1 Kopano's Cell Cache (`cache_cell_size`)

Data that is actually shown to the user in table views, passes through the *cell cache*. This means that any view of a table in Outlook will only retrieve the information from the database of the cells that are not already in the cache. The cache lifetime is as long as the entire server lifetime, so opening an inbox twice in succession should result in 0 disk accesses for the second access. It is a good idea to set the cell cache as high as can be managed, usually about the same size as the MySQL buffer size.

9.2.2 Kopano's object cache (`cache_object_size`)

The Kopano object cache is used to cache the hierarchy table. Each object that is accessed will be placed in this cache, making it faster to retrieve the information again without accessing the database. The more items users have in their folders, the more important this cache becomes. Since the information is quite small, this cache does not need to be large. About 1Mb for 10 users is even an overestimation.

9.2.3 Kopano's indexedobject cache (`cache_indexedobject_size`)

To open a specific item, the program needs to send the server a unique key, called an `entryid`, to the server to request that item. This cache is a 2 way index of the MAPI key to a database key and the other way around. The translation of the keys are quite important. This cache is filled per folder, so large folders will push out otherwise important information. Normal usage is about 0.5 Mb per user.

9.2.4 MySQL `innodb_buffer_pool_size`

The MySQL buffer is used to cache reads and writes to the `ibdata` file. In a dedicated MySQL machine, this would be anywhere between 50% to 80% of the physical RAM size in the machine. When MySQL is run on the same machine as Kopano, it is recommended to be around 25% of physical RAM size (so that Kopano's Cell Cache can also be set to this value)

9.2.5 MySQL `innodb_log_file_size`

The `innodb_log_file_size` is the size of the transaction log. By default there are two logfiles. The preferred value size for the `innodb_log_file_size` is 25% of the `innodb_buffer_pool_size`.

9.2.6 MySQL `innodb_log_buffer_size`

The size of the `innodb_log_buffer_size` that InnoDB uses to write to the log files on disk. A large log buffer allows large transactions to run without a need to write the log to disk before the transactions commit. If big transactions are present, making the log buffer larger will save disk I/O. This value should be 25% of the `innodb_log_file_size`.

9.2.7 MySQL `query_cache_size`

The MySQL query cache is normally disabled. Enabling the query cache can cause a small performance increase, but increasing it to more than a few MBs is not necessary as most recurring SQL queries are rather small.

9.2.8 MySQL `innodb_file_per_table`

The `innodb_file_per_table` option will create per database table a `innodb` data file, instead of using one large `ibdata` file for all data. Having a file per table will give more flexibility to move tables to different filesystem partitions for better performance.

9.2.9 MySQL `max_allowed_packet`

The `max_allowed_packet` defines the maximum size of a single packet which can be inserted in the database. Customer changing this value to a higher value, should keep in mind the Outlook offline database is also using MySQL, which can cause client issues in case packets are larger than 16Mb.

9.3 Setup of modules on different servers

There are several parts of the Kopano server that can be hosted on different servers. In fact, almost each part of the server can be run on a different system. However, in practice, splitting all modules of the server on the different servers, will not increase performance. The main parts that should be considered are:

- *Server1*: MySQL server
- *Server2*: Kopano server
- *Server3*: MTA + AntiSpam/AntiVirus
- *Server4*: WebServer

If these 4 parts were to be hosted on 4 servers, each server would communicate with the others to work as a single system. This setup can be made quite easily simply by configuring the various parts of the system to communicate with another server.

For the MySQL server, this only has to be accessed by the `kopano-server` process on *Server2*. This can very easily be done by setting the correct login and host configuration in Kopano's `server.cfg`.

The Kopano Server will itself be contacted by Outlook Clients, *Server3* (MTA), and *Server4* (WebServer). This can be done because the `kopano-server` process is listening on port 236 on *Server2*, and the other servers can connect with it.

Server3 will accept email on port 25 or fetch email via some email protocol like POP3. After passing the email through anti-spam and anti-virus, the email will be passed to the `kopano-dagent` process. The `kopano-dagent` process can be configured to connect with an SSL certificate with *Server2*. This SSL certificate is required because the `kopano-dagent` needs to be authenticated because it is connecting from a different server over port 236. When this is configured in both *Server3* and *Server2*, the email can be delivered directly to *Server2* by *Server3*.

Server4 is the WebApp server, running Apache, and accepting connections on port 80 (or 443 for SSL). The Kopano WebApp can be configured (in `config.php`) to connect over port 236 (or port 237 for SSL) to *Server2* for the actual data. Once this has been configured, this server is ready to serve users. No additional configuration is required.

Currently, Kopano provides three ways of restoring items:

- Through the softdelete restore system
- Using the brick-level backup system
- With a full database backup (coupled with a backup of the attachment directory)

10.1 Softdelete restore

The softdelete restore can be used by MAPI clients with the *Restore deleted items* dialog to restore deleted items such as Kopano WebApp. This covers most accidental deletions and allows users to directly restore without interaction or supervision from IT administration.

Items that are deleted by the user (by emptying the deleted items folder) are simply placed in the deleted items cache. This means that the item will not actually be removed from the database until the retention time of the item has expired. This expiration time in can be specified in the `server.cfg` configuration time and it set to 30 days by default.

In the following overview, which possibilities can be performed by whom, and when it is most likely used, can be seen.

Table 10.1. Recovery options

Restore request	% of time spent	Backup solution	Performer
Items < 30 days old	80 %	Softdelete system	User and Administrator
Items >= 30 days old	10 %	Bricklevel	Administrator
Items from a specific sender	5 %	Bricklevel	Administrator
Items over a specific time period	3 %	Bricklevel	Administrator
Disaster recovery	2 %	MySQL Dump + Attachments	Administrator

As can be seen, the most common restore request can be performed by the user itself.

When older messages are requested to be restored, the Administrator will need to consult alternatives to the softdelete backup method. It is not possible to restore a single item with a MySQL dump, so this is the point where the `kopano-backup` tool steps in.

The bricklevel backups from the `kopano-backup` tool contain not enough information for disaster recovery. A complete dump of the MySQL database will be needed to perform this type of recovery.

10.2 Full database dump

All the data that is stored by Kopano Server is stored within a MySQL database and on the filesystem for attachments. This means that for a disaster recovery, all that is needed is a full backup/restore of the database in question together with the attachments configured to be stored within `server.cfg`. This can be done in many ways, but we will explain two ways of doing a good backup here. There are certain recommendations on how to make the backup to safely get all data in the state required for a consistent restore.

10.2.1 SQL dump through mysqldump

The contents of an entire Kopano database can be saved to a file by using the `mysqldump` command. There are, however, some options that are important in this case: the `--single-transaction` option should always be specified to `mysqldump`. When this is done, it will cause `mysqldump` to write a single snapshot of the database to disk. This will make sure that any writes done in the database during the backup will not be backed up. In effect, the dump that is made is a ‘snapshot’ of the database at the moment that the dump started. Additionally the option `--routines` should be specified to backup Kopano’s stored procedures.

When using `mysqldump`, it is very important not to do any table locking. This means that the `--opt` option and `--lock-tables` should never be used while dumping a Kopano database. The reason is that these options will ‘lock’ the tables while they are being dumped to disk, causing any accesses to the database to ‘freeze’ while the backup runs. This is firstly unnecessary and secondly may cause emails that are arriving during backup to bounce (depending on the MTA settings).

A simple:

```
mysqldump -p --single-transaction --routines <database> -r <dumpfile>.sql
```

will make a consistent dump of the database.

10.2.2 Binary data dump via LVM Snapshotting

This technique uses the ‘LVM Snapshot’ feature to effectively ‘freeze’ a binary view of the database file, while the database keeps running. This ‘frozen’ view is then simply binary copied to a remote server. This works because `innodb` makes sure that a single snapshot of a database will always be coherent (ie. It will be able to recover the database when `mysql` is started up on this dataset.)

As setting up LVM and configuring LVM for snapshots is a complex process, we refer the user to the LVM documentation and tools on how to set up an LVM volume for the MySQL data, and how to create and delete snapshot partitions.

10.2.3 Attachments backup

When using the attachments storage outside the database, make sure that these attachments are also backed-up.

Some backup methods that can be used to backup the attachments:

- Rsync
- Copy all files to external backup server or external attached hard-drive
- Use of a (commercial) backup agent for Linux, like SEP, Bacula, Arkeia or others

10.3 Brick-level backups

kopano-backup is the brick-level backup tool for Kopano Core. This tool will write a backup of users and stores onto the local filesystem. Currently *kopano-backup* supports three modes of operation:

- initial backup - which includes emails, calendar items, contacts, etc., as well as user settings such as rules, permissions and WebApp settings.
- incremental backup - which is automatically chosen, when the backup directory already contains a backup set for a given user/store.
- purge - which removes items from the backup data, that have previously been deleted from the users store.

For additional options *kopano-backup* includes a man page and a built-in help text, these can be retrieved by calling *man kopano-backup* and *kopano-backup -help*.

In compared to the old way of storing backups in *zarafa-backup*, *kopano-backup* now uses a Berkeley DB to store serialised data and a folder hierarchy resembling the mailbox layout, instead of a binary file with position markers in an index file. As result data exported by *kopano-backup* is now more stable and less prone to corruption. A downside of this new approach is that by updating the Berkeley DB file instead of adding a file for each increment, backups of the exported data are not very efficient. Until [KC-627](#) has been implemented we recommend the use of filesystem snapshots if a versioned backup of the data exported by *kopano-backup* is needed.

10.3.1 kopano-backup vs. zarafa-backup

A brief comparison of usage differences in usage between *kopano-backup* and *zarafa-backup*.

Task	<i>kopano-backup</i>	<i>zarafa-backup</i>
Refresh backup through deleting backup and all increments	not needed anymore as all transactions are recorded via ics and stored in Berkeley DB	should be done on a regular base
Storage of data	Folder hierarchy resembling the mailbox layout, serialised data in Berkeley DB files	One binary file and an index file, increments into separate binary files
off-site backup and rotation	Until KC-627 no efficient way to only transfer increments, filesystem snapshots should be used	possible, while creating increments previous index files should be kept

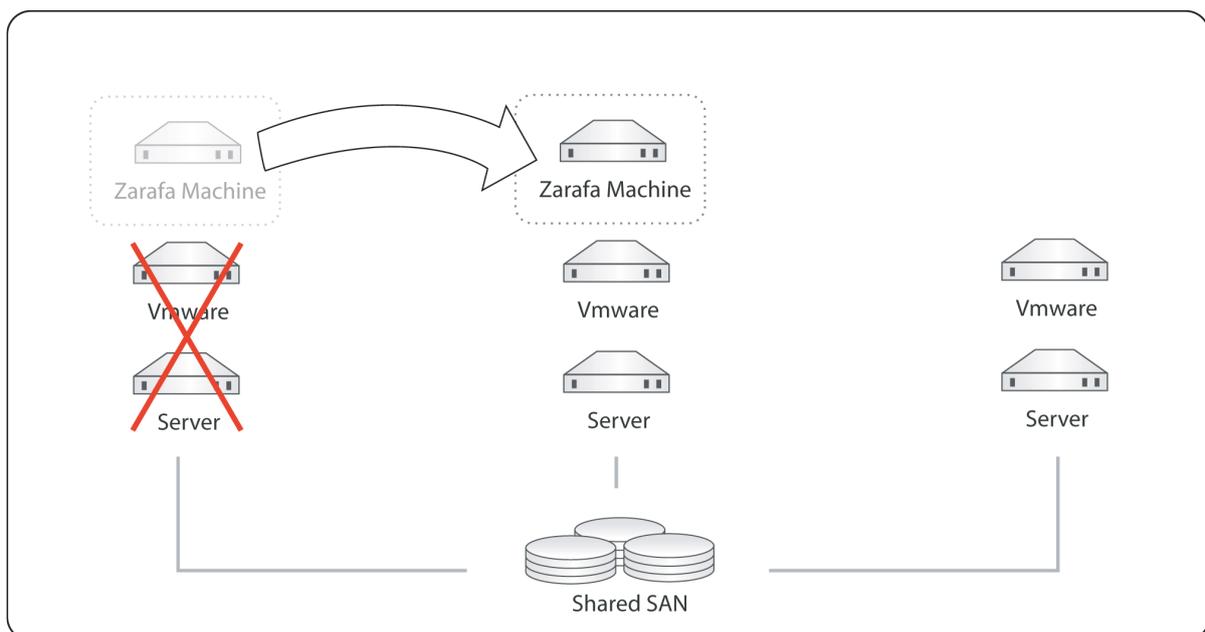
Please note that for disaster recovery it is always recommended to use a MySQL dump over the bricklevel data.

Nowadays the email system is one of the most critical systems within organisations, therefore making the email system high available is getting more important.

The flexible architecture of KC offers different solutions to create a high available mail solution. This whitepaper will show some examples of high available Kopano setups and contains a full description of configuring a High Availability setup with Pacemaker and DRBD.

11.1 High Availability example setups

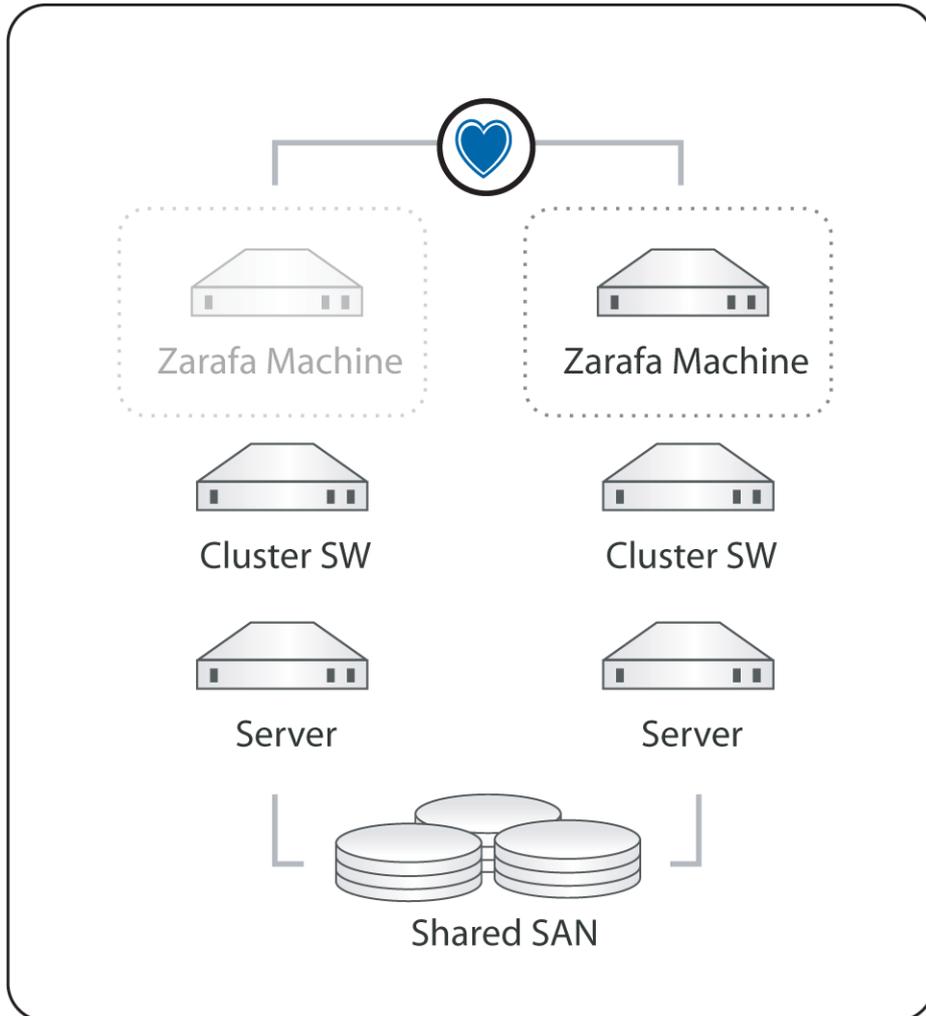
More and more organization will virtualize their server environment to have a limit resource usage and have more flexibility. Most virtualization solutions like VMware Vsphere, Red Hat Enterprise Virtualization, OpenStack and Citrix Xen server will offer high availability as one of the standard features. The HA feature of the virtualization software can also be used for KC. When a hardware failure occurs the virtualization software will automatically start the virtual machine on one of the other virtualization hosts, see figure 1.1.



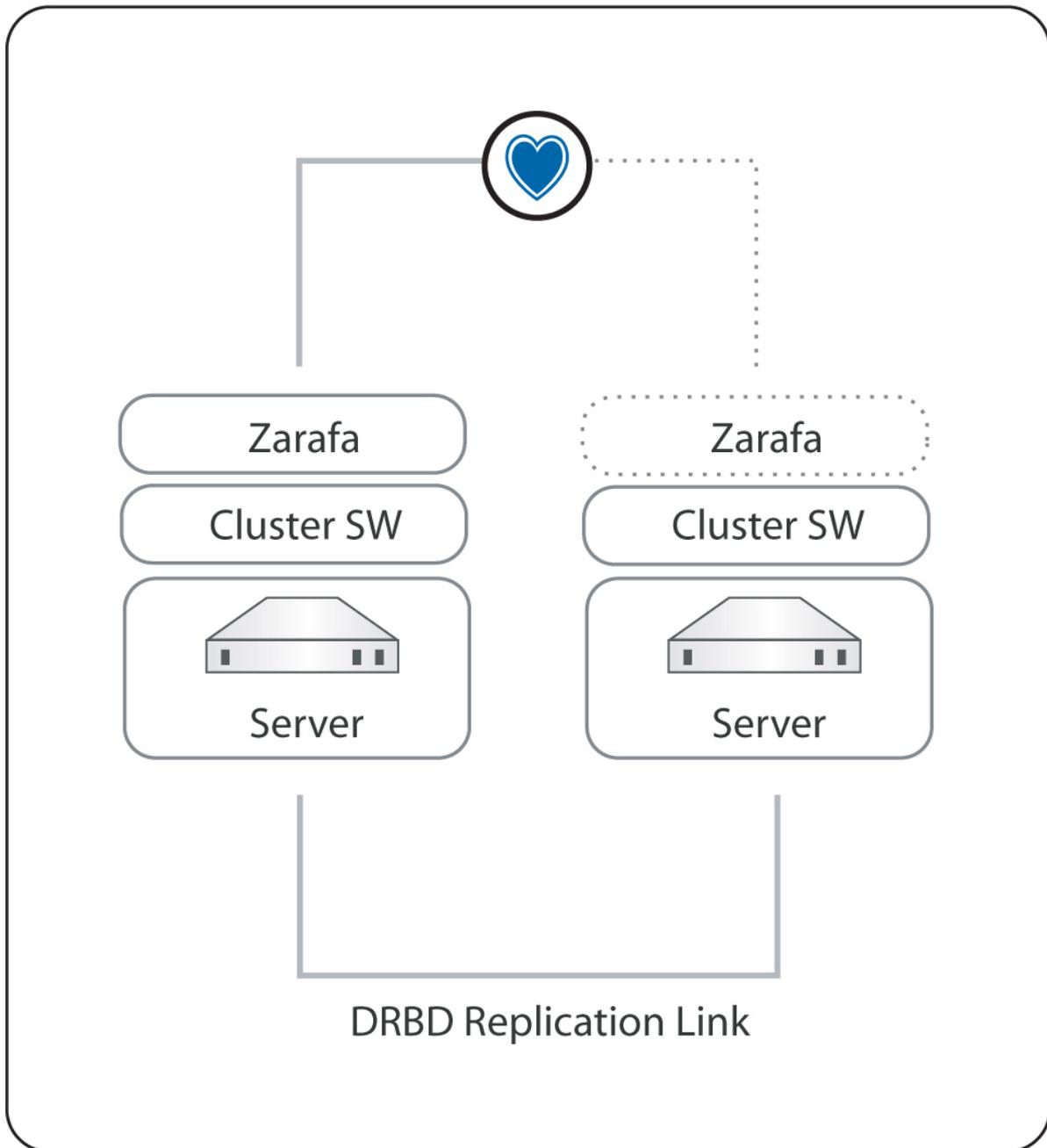
Kopano in a high available virtualization platform

When an organization doesn't have a HA virtualization solution or want to run KC on bare metal to have the best performance, KC can be integrated with different opensource cluster suite solutions, like Red Hat Cluster Suite or Pacemaker.

“Kopano in a high availability setup with a shared storage” shows a High Availability setup where both the MySQL database as the attachments are stored on a shared storage. In case of a failure of one of nodes the resources will be automatically started on the second node. When no shared storage is available the MySQL database and attachments can be stored on a DRBD device (Distributed Replicated Block Device), to have the data available on both nodes. In case of a node failure the DRBD device will be mounted on the second node and the Kopano services will be automatically started, see “Kopano in a high availability setup with DRBD”.



Kopano in a high availability setup with a shared storage



Kopano in a high availability setup with DRBD

Note: When there is a high available virtualization solution, Kopano recommends to use this solution for making the KC stack high available.

11.2 Installing

In the next chapters the installation and configuration of a High Availability setup with Pacemaker and DRBD is described. Pacemaker is a cluster resource manager which is included in most Linux distributions, like RHEL6, SLES11 and Ubuntu. Pacemaker will manage your cluster services by detecting and recovering from node and resource failures, by using the Heartbeat or Corosync messaging layer.

11.2.1 System Requirements

In this whitepaper a two node cluster setup is created based on RHEL6. These system requirements should be solved, before proceeding with this whitepaper:

- Two servers with RAID1 disk array for OS and RAID10 disk array for data storage
- Two network interfaces per machine

11.2.2 Installation

Do on both machines a minimal RHEL6 installation. The RAID10 disk array for the database and attachment storage should not be configured in the installation wizard.

Network configuration

In this whitepaper the two nodes will get the hostname bob and alice. The nodes will be connected with the first network interface to the LAN with subnet 192.168.122.0/24. The second network interface will be used for the DRBD replication.

Servename	bob	alice
eth0	192.168.122.25	192.168.122.26
eth1	10.0.0.25	10.0.0.26

Change the hostname of the nodes in `/etc/sysconfig/network` and configure the network interfaces in `/etc/sysconfig/network-scripts/ifcfg-ethx`. Add the following lines to the `/etc/hosts` file on both nodes.

```
192.168.122.25  bob
192.168.122.26  alice
```

Restart the network services to activate the changes:

```
service network restart
```

Check if the network configuration is successfully configured by using `ifconfig`.

```
[root@bob ~]# ifconfig
eth0  Link encap:Ethernet  HWaddr 52:54:00:4C:30:83
      inet addr:192.168.122.25  Bcast:192.168.122.255  Mask:255.255.255.0
      inet6 addr: fe80::5054:ff:fe4c:3083/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:149 errors:0 dropped:0 overruns:0 frame:0
      TX packets:65 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:12522 (12.2 KiB)  TX bytes:8736 (8.5 KiB)

eth1  Link encap:Ethernet  HWaddr 52:54:00:5F:6F:33
      inet addr:10.0.0.25  Bcast:10.0.0.255  Mask:255.255.255.0
      inet6 addr: fe80::5054:ff:fe5f:6f33/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:27 errors:0 dropped:0 overruns:0 frame:0
      TX packets:29 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:1242 (1.2 KiB)  TX bytes:1530 (1.4 KiB)
      Interrupt:10 Base address:0x6000
```

Package installation

After the network is successfully configured, install and configure KC like described in the Administrator Manual. By default the Kopano services will be started on boot time. In a clustered setup the services will be automatically started by the cluster software, so the Kopano services should be disabled at boot time.

```
chkconfig mysqld off
chkconfig kopano-server off
chkconfig kopano-spooler off
chkconfig kopano-dagent off
chkconfig kopano-gateway off
chkconfig kopano-ical off
chkconfig kopano-monitor off
```

Install the Pacemaker cluster software from the Red Hat yum repository.

```
yum install pacemaker corosync
```

Note: To install the pacemaker software, please make sure you have a valid subscription for the Red Hat High Availability software channel.

The DRBD software is not part of the standard Red Hat repositories. ELRepo.org provides packages, alternatively you can get packages directly from the CentOS Extras repository. Install the drbd packages and the correct drbd kernel module. To find out which kernel is in use, run `uname -a`. For other platforms than RHEL / CentOS, please check out <https://www.linbit.com/en/drbd-community/>

```
rpm -Uvh drbd-8.3.8.1-30.el6.x86_64.rpm drbd-kmdl-2.6.32-71.18.1.el6.x86_64-8.3.8.
↪1-30.el6.x86_64.rpm
```

Enable Corosync and disable DRBD at boot time.

```
chkconfig drbd off
chkconfig corosync on
```

11.2.3 Cluster configuration

Corosync configuration

The communication between the different cluster nodes will be handled by the Corosync software. Execute the following steps to configure Corosync on both nodes:

```
cd /etc/corosync
cp corosync.conf.example corosync.conf
```

Change the `bindnetaddr` in the `corosync.conf` to the local LAN subnet address.

```
bindnetaddr: 10.0.0.0
```

To instruct CoroSync to start Pacemaker, create `/etc/corosync/service.d/pcmk` with the following fragment.

```
service {
    # Load the Pacemaker Cluster Resource Manager
    name: pacemaker
    ver: 0
}
```

Restart Corosync to activate the changes.

```
service corosync restart
```

11.3 DRBD device initialization

In order to have the MySQL database and the attachments on both nodes available, two DRBD devices will be created. Each DRBD device needs on both nodes a partition on the RAID10 device.

Create on both nodes two partitions on the RAID10 device. In this whitepaper the RAID10 device is available as /dev/sdb.

```
fdisk /dev/sdb
```

Use the following steps to initialize the partitions. .

```
Command (m for help): n
Command action
    e   extended
    p   primary partition (1-4)
    p
Partition number (1-4): 1
First cylinder (1-2031, default 1):
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-2031, default 2031): 100G

Command (m for help): n
Command action
    e   extended
    p   primary partition (1-4)
    p
Partition number (1-4): 2
First cylinder (501-2031, default 501):
Using default value 501
Last cylinder, +cylinders or +size{K,M,G} (501-2031, default 2031):
Using default value 2031

Command (m for help): w
The partition table has been altered!
```

The partitions can now be used as DRBD devices. Add the following DRBD configuration to /etc/drbd.conf on both nodes:

```
global {
    usage-count no;
}

common {
    protocol C;
    syncer {
        rate 50M;
    }
}

resource mysql {
    on bob {
        device /dev/drbd0;
        disk /dev/sdb1;
        address 10.0.0.25:7788;
        meta-disk internal;
    }
    on alice {
```

```

        device /dev/drbd0;
        disk /dev/sdb1;
        address 10.0.0.26:7788;
        meta-disk internal;
    }
}

resource kopano {
    on bob {
        device /dev/drbd1;
        disk /dev/sdb2;
        address 10.0.0.25:7799;
        meta-disk internal;
    }
    on alice {
        device /dev/drbd1;
        disk /dev/sdb2;
        address 10.0.0.26:7799;
        meta-disk internal;
    }
}

```

Reload DRBD on both nodes to activate the changes.

```
service drbd reload
```

Before the DRBD devices can be used, both resources has be initialized. Run the following commands on both nodes.

```
[root@bob etc]# drbdadm create-md mysql
Writing meta data...
initializing activity log
NOT initialized bitmap
New drbd meta data block successfully created.
```

```
drbdadm up mysql
```

```
drbdadm create-md kopano
Writing meta data...
initializing activity log
NOT initialized bitmap
New drbd meta data block successfully created.
```

```
drbdadm up kopano
```

Check if the DRBD devices are successfully created, by using the following command:

```
[root@bob etc]# cat /proc/drbd
version: 8.3.8.1 (api:88/proto:86-94)
GIT-hash: 0d8589fcc32c874df57c930ca1691399b55ec893 build by gardner@, 2011-02-23_
↪08:32:21
0: cs:Connected ro:Secondary/Secondary ds:Inconsistent/Inconsistent C r----
   ns:0 nr:0 dw:0 dr:0 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:b oos:251924
1: cs:Connected ro:Secondary/Secondary ds:Inconsistent/Inconsistent C r----
   ns:0 nr:0 dw:0 dr:0 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:b oos:771564
```

The DRBD synchronisation can be start with the following command on bob.

```
[root@bob ~]# drbdadm -- --overwrite-data-of-peer primary all
```

To check the progress of the synchronisation, use `cat /proc/drbd`.

```
[root@bob ~]# cat /proc/drbd
version: 8.3.8.1 (api:88/proto:86-94)
GIT-hash: 0d8589fcc32c874df57c930ca1691399b55ec893 build by gardner@, 2011-02-23_
↪08:32:21
0: cs:SyncSource ro:Primary/Secondary ds:UpToDate/Inconsistent C r----
ns:94336 nr:0 dw:0 dr:103160 al:0 bm:5 lo:2 pe:87 ua:256 ap:0 ep:1 wo:b oos:160340
  [=====>.....] sync'ed: 37.1% (160340/251924)K
  finish: 0:00:29 speed: 5,328 (5,088) K/sec
```

Both DRBD devices can now be formatted with a filesystem.

```
[root@bob ~] mkfs.ext4 /dev/drbd0
[root@bob ~] mkfs.ext4 /dev/drbd1
```

11.4 Pacemaker configuration

Before the actual cluster configuration can be done, the mysql and kopano service will be assigned to a cluster ip-address. The cluster ip-addresses which are used in this example are:

```
mysql 192.168.122.101
kopano 192.168.122.100
```

Add to the file `/etc/my.cnf` the bind-address in the `[mysqld]` section. Make sure this change is done on both nodes.

```
bind-address = 192.168.122.101
```

To let the kopano-server access the MySQL database, the privileges has to be set.

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'root'@'192.168.122.0/255.255.255.0'
↪IDENTIFIED BY 'secret';
mysql> FLUSH PRIVILEGES;
```

Change in the file `/etc/kopano/server.cfg` the `server_bind` to 192.168.122.100.

```
server_bind = 192.168.122.100
```

The kopano-server will connect to the cluster ip-address of MySQL. Make sure the `mysql_host` in `/etc/kopano/server.cfg` is correctly set.

```
mysql_host = 192.168.122.101
```

The kopano-dagent should also listen on the kopano cluster ip-address, so the Postfix MTA's on both nodes can deliver emails. Change in the file `/etc/kopano/dagent.cfg` the `server_bind` address to 192.168.122.100.

```
server_bind = 192.168.122.100
```

Change in the file `/etc/postfix/main.cf` the `virtual_transport` to the cluster ip-address instead of localhost. The Postfix service itself, will not be part of the cluster services.

```
virtual_transport = lmtp:192.168.122.10:2003
```

When the kopano-gateway and kopano-ical will be used, the `server_socket` of this processes should be changed. Change in `/etc/kopano/gateway.cfg` and `/etc/kopano/ical.cfg` the `server_socket`.

```
server_socket = http://192.168.122.100:236/
```

The Pacemaker cluster configuration can now be done.

The Pacemaker cluster suite offers different tools to configure the cluster configuration. Some Linux distributions, like SLES11 include a graphical administration interface, but RHEL6 is not including this interface at the moment. Another tool for configuring the cluster is the CLI tool, called `crm`.

This tool will be used to configure for this cluster setup and to manage both nodes and resources. More information about the `crm cli` can be found in the [Pacemaker documentation](#).

First the cluster will be changed to disable automatic fencing and quorum support for this two node cluster.

```
crm configure property stonith-enabled=false
crm configure property no-quorum-policy=ignore
```

The resources can now be configured. Two resource groups will be defined in this cluster, one group for MySQL and one for all kopano services. A resource group will contain the following steps:

1. Make the DRBD resource primary
2. Mount the DRBD device
3. Assign cluster ip-address
4. Start the services

Execute the following commands to add the mysql resources.

```
crm(live)# configure
crm(live)# edit
    primitive drbd_mysql ocf:linbit:drbd \
    params drbd_resource="mysql" \
    op start interval="0" timeout="240" \
    op stop interval="0" timeout="100" \
    op monitor role=Master interval=59s timeout=30s \
    op monitor role=Slave interval=60s timeout=30s
    primitive mysql_fs ocf:heartbeat:Filesystem \
    params device="/dev/drbd0" directory="/var/lib/mysql" fstype="ext4"
    <--options="noatime" \
    op monitor interval="30s"
    primitive mysql_ip ocf:heartbeat:IPaddr2 \
    params ip="192.168.122.101" cidr_netmask="32" nic="eth0" \
    op monitor interval="30s"
    primitive mysqld lsb:mysqld \
    op monitor interval="10" timeout="30" \
    op start interval="0" timeout="120" \
    op stop interval="0" timeout="120"
    group mysql mysql_fs mysql_ip mysqld
    ms ms_drbd_mysql drbd_mysql \
    meta master-max="1" master-node-max="1" clone-max="2" clone-node-max="1"
    <--" notify="true"
    colocation mysql_on_drbd inf: mysql ms_drbd_mysql:Master
    order mysql_after_drbd inf: ms_drbd_mysql:promote mysql:start

crm(live)# commit
```

The mysql resources are now configured, to check the status of the resources use:

```
crm(live)# status
=====
Last updated: Sun Feb 27 22:42:20 2011
Stack: openais
Current DC: alice - partition with quorum
Version: 1.1.2-f059ec7ced7a86f18e5490b67ebf4a0b963bccfe
2 Nodes configured, 2 expected votes
2 Resources configured.
=====
```

```

Online: [ bob alice ]

Resource Group: mysql
  mysql_fs (ocf::heartbeat:Filesystem): Started bob
  mysql_ip (ocf::heartbeat:IPAddr2): Started bob
  mysqld (lsb:mysqld): Started bob
Master/Slave Set: ms_drbd_mysql
  Masters: [ bob ]
  Slaves: [ alice ]

```

Now the Kopano resource group can be added.

```

crm(live) # configure
crm(live) # edit
  primitive drbd_kopano ocf:linbit:drbd \
  params drbd_resource="kopano" \
  op monitor interval="60s"
  primitive kopano_fs ocf:heartbeat:Filesystem \
  params device="/dev/drbd1" directory="/var/lib/kopano" fstype="ext4" \
  op start interval="0" timeout="240" \
  op stop interval="0" timeout="100" \
  op monitor role=Master interval=59s timeout=30s \
  op monitor role=Slave interval=60s timeout=30s
  primitive kopano_ip ocf:heartbeat:IPAddr2 \
  params ip="192.168.122.100" cidr_netmask="32" nic="eth0" \
  op monitor interval="30s"
  primitive kopano-server lsb:kopano-server \
  op monitor interval="30" timeout="60"
  primitive kopano-dagent lsb:kopano-dagent \
  op monitor interval="30" timeout="30"
  primitive kopano-gateway lsb:kopano-gateway \
  op monitor interval="30" timeout="30"
  primitive kopano-ical lsb:kopano-ical \
  op monitor interval="30" timeout="30"
  primitive kopano-monitor lsb:kopano-monitor \
  op monitor interval="30" timeout="30"
  primitive kopano-spooler lsb:kopano-spooler \
  op monitor interval="30" timeout="30"
  group kopano kopano_fs kopano_ip kopano-server \
  kopano-spooler kopano-dagent kopano-monitor kopano-gateway kopano-ical
  ms ms_drbd_kopano drbd_kopano \
  meta master-max="1" master-node-max="1" clone-max="2" clone-node-max="1"
  <notify="true"
  colocation kopano_on_drbd inf: kopano ms_drbd_kopano:Master
  order kopano_after_drbd inf: ms_drbd_kopano:promote kopano:start
  order kopano_after_mysql inf: mysql:start kopano:start

crm(live) # commit

```

To check the status of the cluster services use:

```

crm(live) # status
=====
Last updated: Mon Feb 28 08:31:32 2011
Stack: openais
Current DC: bob - partition WITHOUT quorum
Version: 1.1.2-f059ec7ced7a86f18e5490b67ebf4a0b963bccfe
2 Nodes configured, 2 expected votes
4 Resources configured.
=====

Online: [ bob ]
OFFLINE: [ alice ]

```

```
Resource Group: mysql
  mysql_fs (ocf::heartbeat:Filesystem): Started bob
  mysql_ip (ocf::heartbeat:IPaddr2): Started bob
  mysqld (lsb:mysqld): Started bob
Master/Slave Set: ms_drbd_mysql
  Masters: [ bob ]
  Stopped: [ alice ]
Resource Group: kopano
  kopano_fs (ocf::heartbeat:Filesystem): Started bob
  kopano_ip (ocf::heartbeat:IPaddr2): Started bob
  kopano-server (lsb:kopano-server): Started bob
  kopano-spooler (lsb:kopano-spooler): Started bob
  kopano-dagent (lsb:kopano-dagent): Started bob
  kopano-monitor (lsb:kopano-monitor): Started bob
  kopano-gateway (lsb:kopano-gateway): Started bob
  kopano-ical (lsb:kopano-ical): Started bob
Master/Slave Set: ms_drbd_kopano
  Masters: [ bob ]
  Stopped: [ alice ]
```

The Apache webserver will be configured to run on both nodes, so a loadbalancer can be placed in front of the nodes. The Apache resource will check the status of the resource, by using the `server-status` page.

The `server-status` should be enabled in the Apache configuration file. Uncomment the following lines in the file `/etc/httpd/conf/httpd.conf`.

```
<Location /server-status>
  SetHandler server-status
  Order deny,allow
  Deny from all
  Allow from 127.0.0.1
</Location>
```

Now the Apache resource can be added to cluster configuration.

```
crm(live)# configure
crm(live)# edit
  primitive apache ocf:heartbeat:apache \
  params configfile="/etc/httpd/conf/httpd.conf" \
  op monitor interval="60s" \
  op start interval="0" timeout="40s" \
  op stop interval="0" timeout="60s"
clone apache_clone apache
crm(live)# commit
```

The Kopano WebApp should connect to the cluster ip-address of Kopano, to be available on both nodes. Change the `server_socket` in `/etc/kopano/webapp/config.php`.

```
define("DEFAULT_SERVER", "http://192.168.122.100:236/");
```

Now the cluster configuration is ready and can be used.

11.5 Testing configuration

Before the cluster will be used for production use, it's important to test different failover scenarios. The tool `crm_mon` will show the realtime status of the cluster.

```
=====
Last updated: Mon Feb 28 18:41:16 2011
```

```

Stack: openais
Current DC: bob - partition with quorum
Version: 1.1.2-f059ec7ced7a86f18e5490b67ebf4a0b963bccfe
2 Nodes configured, 2 expected votes
5 Resources configured.
=====

Online: [ bob alice ]

Resource Group: mysql
    mysql_fs (ocf::heartbeat:Filesystem): Started bob
    mysql_ip (ocf::heartbeat:IPaddr2): Started bob
    mysqld (lsb:mysqld): Started bob
Master/Slave Set: ms_drbd_mysql
    Masters: [ bob ]
    Slaves: [ alice ]
Resource Group: kopano
    kopano_fs (ocf::heartbeat:Filesystem): Started bob
    kopano_ip (ocf::heartbeat:IPaddr2): Started bob
    kopano-server (lsb:kopano-server): Started bob
    kopano-spooler (lsb:kopano-spooler): Started bob
    kopano-dagent (lsb:kopano-dagent): Started bob
    kopano-monitor (lsb:kopano-monitor): Started bob
    kopano-gateway (lsb:kopano-gateway): Started bob
    kopano-ical (lsb:kopano-ical): Started bob
Master/Slave Set: ms_drbd_kopano
    Masters: [ bob ]
    Slaves: [ alice ]
    Clone Set: apache_clone
    Started: [ bob alice ]

```

11.6 Testing a node failure

1. Login to alice and start `crm_mon`
2. Give bob a hard shutdown
3. Check if all services will be successfully started on alice

11.7 Testing a resource failure

1. Login to bob and start `crm_mon`
2. Shutdown the kopano-server with `killall -9 kopano-server`
3. Check if the kopano-server is successfully restarted

Try this test for different services.

11.8 Getting more information

The following links will give more useful information about DRBD, Pacemaker or the `crm` commandline tool.

- <http://www.drbd.org/users-guide> for all documentation about installing, configuring and trouble shooting DRBD
- <http://clusterlabs.org/pacemaker/doc/> for a complete reference of all `crm` commandline interface
- <http://clusterlabs.org> for many different example setups and architecture of Pacemaker

Please see the *Kopano Changelog* <http://documentation.kopano.io/kopano_changelog> for updates changelogs.

12.1 Release notes for 8.5.0 (2018-02-05)

Enhancements:

- server: new “server_listen” directive replacing “server_bind” [KC-645]
- server: stronger keep-alive [KC-888,KC-890]
- server: further general performance improvements [KC-62,KC-181,KC-889,KC-892,KC-893]
- server: update PR_LOCAL_COMMIT_MAX on hard-deletes [KC-770]
- server: speed up contact and search folder querying [KC-265,KC-941]
- server: skip some unnecessary attachment accesses [KC-769,KC-794]
- spooler: introduce indexed_headers config directive [KC-948]
- search: pass “limit_results” to xapian to improve performance [KC-786]
- search: optionally index draft folders [KC-787]
- unixplugin: support multiple non_login_shells [KC-824]
- unixplugin: add /sbin/nologin as a non_login_shell (new installs only) [KC-824]
- gateway: RFC 6154 support [KC-857]
- dagent: a Python version of kopano-autorespond is available [KC-861]
- kopano-spamd: new program [KC-666]
- icalmapi: support URL, NICKNAME, PRODID in vcards
- php: extend mapi_feature with ST_ONLY_WHEN_OOF [KC-970]

Fixes:

- gateway: generate envelope using inetmapi if not present yet [KC-607]
- spooler: only evaluate rules that are explicitly enabled using PR_RULE_STATE [KC-963]
- search: supply a HOME environment (tmpdir) when running conversion tools [KC-331]

Changes:

- Support for Debian 7 ended [KC-736]
- /etc/kopano is no longer prepopulated, create .cfg manually if you need to override anything [KC-681, KC-978]
- server: remove support for upgrading databases older than ZCP 7.2 [KC-839]
- gateway: use threaded mode for reduced memory usage on many-user systems (new installs only) [KC-768]
- gateway: the “imap_store_rfc822” config directive is removed [KC-964]
- server: the “counter_reset” config directive is removed [KC-960]
- spooler: the “always_send_utf8” config directive is removed [KC-901]
- client: MAPI provider configuration moved from /etc/mapi to /usr/lib/mapi.d

Packager notes:

- libical 3.x support [KC-920]

12.2 Release notes for 8.4.7

Fixes:

- php: do return true when AbortSubmit succeeded [KW-2087]

12.3 Release notes for 8.4.6 (2018-02-02)

Fixes:

- common: restore support for binary data in RTF [KC-969]
- libserver: store size for orphaned stores was reported incorrectly [KC-984]
- client: have OpenEntry check for NULL entryids and entryids too short [KC-932]
- dagent, client: fix nonfunctional HTML filter [KC-953]
- common: switch logging to stderr when pipe dies [KC-815]
- spooler: avoid printing garbage when non-worker child exits [KC-815]

12.4 Release notes for 8.4.5 (2017-12-15)

Fixes:

- treewide: avoid freeing ADRLIST garbage pointers [KC-927]
- libserver: fix waiting for ntlm_auth forever [KC-916]
- libserver: fix use after free in ECCacheManager::GetPropFromObject [KC-60, KC-177, KC-355, KC-669, KC-754]

12.5 Release notes for 8.4.4 (2017-11-23)

Fixes:

- common: fix detection of local connections that need not use zlib compression

- libserver: improve ECICS error reporting [KC-880]
- inetmapi: overwrite recipients instead of appending [KC-419]

12.6 Release notes for 8.4.3 (2017-11-07)

Enhancements:

- dagent: enable automated backtraces when invoked with -f [KC-879]

Fixes:

- php5-ext: fix positive retval setting in error case [KC-875]
- dagent: redirect rule led to crash [KC-868,KC-871]

12.7 Release notes for 8.4.2 (2017-11-02)

- server: revert NO_UNSIGNED_SUBTRACTIONS [KC-841,KC-869]

12.8 Release notes for 8.4.1 (2017-11-01)

Fixes:

- Avoid calling srand with 1-second-granular time
- inetmapi: handle empty/invalid Sender in RFC2822 mails [KC-263]
- spooler: for send-later mails, check trash, not outbox [KC-848,KC-863]

12.9 Release notes for 8.4.0 (2017-10-30)

Enhancements:

- dagent, gateway: whitelist-based HTML filter [ZCP-13472]
- New scripts/utilities: kopano-fix-ipm-subtree, kopano-localize-folders, kopano-recreate-systemfolders, kopano-rules [KC-533]
- server: drop excessive locking in ECABObjectTable/ECStoreObjectTable hot path
- provider: speed up getIDsFromNames by reducing SQL queries
- server: add LIMIT clauses to single-result SELECT statements [KC-5]
- client: speedup from-scratch MAPI session creation by avoiding extraneous logon-logoff cycles during provider initialization [KC-667]
- client: add API for dump+restore of MAPI session profile data so libmapi users can skip provider reinitialization at program startup [KC-67,KC-165]
- gateway: add option to ignore commands during IDLE
- gateway: will now warn about IMAP clients using wrong sequence ranges
- php: copy back improvements from the Z-Push project's bundled copy [KC-463]
- search: make searchfolder creation in shared stores configurable [KC-565]
- mapi: disable very slow RTF compression [KC-622]
- server: add entry cache for S3 backend [KC-702]

- icalmapi: handle up to three email addresses in a vcard [KC-420]
- inetmapi: add the right extension for attachments without filename [KC-624]
- search: index embedded messages (recursively) [KC-151]
- migration-pst: new options -S, -clean-folders option [KC-651,KC-485]
- pyko: fall back to search-key to determine recipient email address [KC-566]
- pyko: support processing basic cancellations [KC-612]
- pyko: support cancellation of existing exception [KC-612]
- icalmapi: support ADR, ORG, TITLE tags in VCF files
- swig: lazy opening of folder objects [KC-632]
- backup: the “backup_servers” option is back again [KC-364]
- backup: save and restore store-level ACLs [KC-687]
- backup: use store GUID for backup directory [KC-686]
- backup: merge store-level metadata [KC-627]

Fixes:

- spooler: do not run mr-process before mr-accept [KC-498]
- spooler: do not send deleted send-later mails [KC-848]
- Coverity reports on absent return value checks [KC-595]
- freebusy: avoid potential division by zero when trying to determine the end date of a non-recurring recurrence. [KC-595]
- libserver: ensure same endianness for SOURCEKEYs [KC-628]
- server plugins: more escaping in SQL commands [KC-620]
- gateway: set PR_FOLLOWUP_ICON for WebApp [KC-653]
- gateway: cure slow folder access [KC-853]
- gateway: resolve accessing invalid pointer [KC-817]
- server: avoid crash when NTLM subprocess gives no newline [KC-656]
- pyko: avoid hang on shutdown of Python services with logging [KC-643]
- server: disable reminders from shared stores [KC-728]
- server: improved error reporting when users/groups/group members are not found/not complete [KC-497]
- backup: avoid emitting tracebacks [KC-411]

Changes:

- server: the underlying call for traditional-style fd monitoring was changed from select(2) to poll(2) [ZCP-13065]
- server: the sync_log_all_changes is obsolete [KC-527]
- server: use SQL autocommit=0 during transactions
- server: stop relying on NO_UNSIGNED_SUBTRACTIONS and avoid use of hexadecimal numbers [KC-841]
- server: avoid UB during hostname lookup when host is IPv6-only
- gateway: ignore missing (lost) attachments [KC-363]
- dagent: turn on PHP7 SCL on RHEL6 [KC-621]
- pyko/backup: change error into warning for ‘missing’ attachments [KC-545, KC-555, KC-575]

- server: hide private messages' reminders from shared stores [KC-565]
- server: drop ZCP client update support [KC-644]

Packager notes:

- libicu changed from optional to required build dependency
- kopano-server S3 store required libs3 4.1 [KC-751]
- python/swig is now optional [KC-753]

12.10 Release notes for 8.3.5 (unreleased/state of 2017-10-31)

Fixes:

- server: complete signal blocking [KC-779]
- ical: avoid freeing garbage pointers [KC-803]
- caldav: fix unbounded copy/iteration past end [KC-792]
- migration-pst: skip archiver properties [KC-812]
- icalmapi: cure a NULL dereference when generating VCFs
- inetmapi: cure a NULL dereference when parsing MDNs [KC-814]
- inetmapi: support embedded messages once again [KC-540, KC-775]
- inetmapi: stop generating <"@"@hostname> [KC-689, KC-772]
- server: admin user is to always open shared reminders [KC-813]
- gateway: fix infinite loop when UID ranges are inverted [KS-38641]
- extra nullptr checks in Util::HrCopyProperty [KC-826]
- server: removing user from folder permission broke [KC-844]
- ldapplugin: fix out-of-bounds reads in SMD5 and SSHA password check
- ldapplugin: avoid triggering crash in DES_encrypt
- ldapplugin: fix truncated SMD5 hash comparison
- ldapplugin: fix out-of-bounds in b64_encode

12.11 Release notes for 8.3.4 (2017-09-01)

Fixes:

- server: configurable shared reminders [KC-789]
- migration-pst: create unknown named-properties [KC-788]
- server: filter private shared messages from search, and filter private shared notifications [KC-565]
- dagent: check quota on delivery
- m4lcommon: ensure right condition for SRowSetPtr::empty [KC-773]

Changes:

- server: rename disable_shared_reminders option [KC-565]

12.12 Release notes for 8.3.3 (2017-08-09)

Fixes:

- gateway: fix another IMAP protocol error [KC-720]
- server: fix depth level count for attachments [KC-745]
- common/spooler: catch a potential pointer underflow [KC-694]
- common: fix incorrect condition in scheduler [KC-638] Fixes timing of softdeletes, sync-table cleanups, kopano-monitor.
- server: disable reminders from shared stores [KC-728]
- libserver: avoid creating multi-stream gzip files [KC-104, KC-314, KC-597]
- monitor: do not check quota for admin user [KC-773]

Changes:

- server: build with libs3 4.1 [KC-751]

12.13 Release notes for 8.3.2 [2017-07-06]

Fixes:

- gateway: fix an IMAP protocol error [KC-668] Apple Mail/Alpine did not show mails with long encoded subjects
- common: restore ability to output crashdump [KC-630]
- caldav: counter proposal without dates [KC-710]
- inetmapi: avoid short allocation on group expansion [KC-388,KC-727]
- server: support ICS initial sync with more than 100000 items again [KC-683]

Changes:

- server: improve wording of ntlm_auth messages [KC-572]

12.14 Release notes for 8.3.1 [2017-06-20]

Enhancements:

- gateway: add option to ignore commands during IDLE

Fixes:

- dagent.cfg:forward_whitelist_domains commentary has been reworded [KC-593]
- inetmapi: do not force HTML when use_tnef is set to minimal [KC-664]
- spooler: avoid a use-after-free, and a deadlock after this failure [KC-588]
- backup: batch storage updates [KC-662]
- spooler: custom bounce message text for forward_whitelist_domain [KC-618]
- server: avoid unchecked return value and unsigned underflow [KC-656]
- php: rework pointer value storing
- daemons: call initgroups when switching user and don't fall over [KC-684,KC-690]

12.15 Release notes for 8.3.0 [2017-04-27]

Enhancements:

- mapi: drop global lock and replace singleton allocmore table by per-object vectors [KC-328]
- swig: expose group and company properties in Python [KC-320]
- xapiant-compact.py: new -c option to specify config file [KC-205]
- utils: support setting out-of-office without an until-date [KC-275]
- gateway, server: reload SSL certificates on SIGHUP [KC-301]
- gateway: optimize LIST, SELECT, STATUS [KC-490]
- dagent: log_raw_message option can now be used selectively on users [KC-370]
- icalmapi: VCF conversion [KC-420]
- migration-pst: call SaveChanges only once [KC-534]

Fixes:

- pyko: do not throw backtraces on log messages [KC-340]
- server: Ctrl-C now works in gdb [KC-171]
- ics: make creation of new syncids work incrementally [KC-208]
- libserver: change incorrect compare operator for EID_V0 [KC-365]
- migration-pst: show usage, not traceback, for invalid options [KC-372]
- migration-pst: skip root folder more intelligently [KC-487]
- migration-pst: MV properties are handled better [KC-457]
- inetmapi: avoid buffer overread on rejected recipients (showed garbage in logs) [KC-398]
- client: add extra checks for EID sizes [KC-500]
- gateway: enforce user and password checking on local socket [KC-396,KC-490]
- caldav: avoid a nullptr dereference [KC-236]
- cachestat: avoid exception and unpack tuple [KC-402]
- ldapplugin: revert “catch empty ldap_search_base” [KC-602]
- spooler: fix crash on forwarding rules [KC-608]
- dagent: avoid shell command injections [KC-619]
- server: avoid returning garbage for getLicenseAuth [KA-2]

Changes:

- server: compressed attachments now get the same permissions as uncompressed ones [KC-380]
- server: make softdelete_lifetime config setting a reloadable property [KC-472]
- icalmapi: handle missing timezone for RRULE [KC-414]
- backup: maintain deleted folders and add -purge N option [KC-376]
- migration-pst: filter metadata at start of subject [KC-424]
- migration-pst: ignore decode errors [KC-521]
- common: fix empty text bodies when converting U+0000 from RTF/HTML [KC-557,KC-580]
- icalmapi: reworked copying description into mail body [KC-568]

Of special mention:

- search: python3 support (but requires new python-xapian and, as a result, a db migration or full reindexing)

Developer/packager notes:

- KC variables and functions now live in the KC:: C++ namespace [KC-369]
- Build-time requirements: gsoap >= 2.8.39 [KC-335], libvmime >= 0.9.2, boost no longer needed at all [KC-451], xmlto no longer needed at all.

12.16 Release notes for 8.2.0 [2017-02-17]

Fixes:

- backup: avoid exceptions on problematic rules/ACLs/delegates [KC-213,KC-266]
- The comment for server.cfg's "disabled_features" was wrong [KC-262]
- php: fix crash by adding missing pointer type conversions [KC-274]
- dagent: the "Received" debugging header had the wrong target address
- gateway: do not emit an X-Mailer field when retrieving mail [KC-277]
- gateway: report missing attachments over IMAP better [KC-436]
- server/ldap: report empty ldap_search_base setting
- client: verify peer's SSL certificate name [KC-156,KC-409]
- admin: support unwrapping "default:" type URLs [KC-289]
- backup: fix tracebacks when used with ZCP [KC-306,KC-307,KC-308]
- server: implement missing readback of compressed attachments [KC-285]
- dagent: iCal descriptions caused wrong body parts to be displayed [KC-138]
- dagent: mr-process failed to copy attachments to the calendar item [KC-202]
- dagent: restore/rework forced ASCII charset upgrade [KC-294]
- server: S3 attachment size was not calculated [KC-351]
- inetmapi: review misdetection of TNEF necessity for reminders [KC-348]
- icalmapi: unbreak timezone lookup [KC-313]
- icalmapi: handle RRULE with missing timezone [KC-341]
- inetmapi: avoid an infinite recursion on SMIME handling [KC-366]
- inetmapi: avoid buffer overread when generating NDR [KC-398]
- inetmapi: avoid overzealously generating winmail.dat [KC-348]
- server: equalize fs permissions for attachments [KC-380]
- migration-pst: resolve tracebacks [KC-372,KC-373,KC-377]
- migration-pst: do not skip folder when items unprocessible [KC-417]
- common: fix spurious crash in sk_SSL_COMP_free on shutdown [KC-443]

Enhancements:

- client: Kerberos/GSSAPI support [KC-396]
- PST importer [KC-59]
- Python 3 support [KC-48,KC-267]
- search: files are now compacted, and their uid/gid checked [KC-188]

- server: allow search folder creation outside of own store [KC-271]
- dagent: forwarding by rule can be restricted with a whitelist [KC-109]
- search: add script for findroot upgrade [KC-300]
- php: can build with ZTS again [KC-442]
- php: ICS import/export functions [KC-302]
- server: AWS4-HMAC-SHA256 support for S3 [KC-170]
- pyko: permit “public@company” syntax to specify stores [KC-317]
- dagent: new AUTORESPOND_BCC option for use with OOF [KC-319]
- kopano-stats: bind ‘q’ key to exit as well [KC-105]
- presence: log authentication errors
- Improved PHP7 support [*,KC-330]
- backup: backup deleted items and folders [KC-376]
- backup: add –purge option [KC-376]
- backup: improved logging when ACL does not resolve to user/group [KC-431]

Changes:

- Non-Delivery Reports now originate from “Mail Delivery System” (like postfix) instead of yourself [KC-309]
- Support for building with a no-SSLv2 OpenSSL 1.1. [KC-230] If you run such a setup, be aware that a config setting like “ssl_protocol = !SSLv2” in one or more of kopano-{server,gateway,ical}.cfg can inhibit the process from starting.
- Cleanup of the example LDAP configuration files. [KC-229] /usr/share/doc/kopano/example-configs/ now has just a ldap.cfg, and no more ldap{,ms}.{active-directory,ldap}.cfg.
- The example LDAP config file now has a different proposed value for ldap_object_search_filter for OpenLDAP. [KC-218]
- spooler: messages with reminder will be sent with a TNEF copy [KC-152]
- admin: group features will no longer be shown [KC-239]
- search: log to file (if set) instead of stdout [KC-204]
- search: treat ‘_’ as a word break [KC-290]
- swig: resolve crash when python programs end [KC-269]
- config: change ldap_object_search_filter for WebApp to be able to search by mail address [KC-337]
- backup: synchronize soft-deleted items [KC-376]
- The RTF encoder incorrectly produced paragraphs where it should have created linefeeds [KC-338]
- The RTF decoder failed to see that uXXXX could start a paragraph [KC-338]
- The RTF decoder erroneously created a new paragraph on pard [KC-338]

Developer/packager notes:

- Support for building the source with newer gsoap (upto and including 2.8.37) [KC-261]
- KC 8.2 is the first to support 2.8.34+ at runtime. [KC-261] Earlier KC releases only support gsoap < 2.8.30 because KC was using undocumented behavior for which the generator changed the wire protocol.
- New libvmime API is now being used [KC-263]

Internal:

- Many “goto exit” were abolished and reordered [KC-87]

- for() loop verbosity reduced by using range-based loops [KC-88]
- pthread_mutex calls have been switched to std::mutex [KC-191]
- Coverity report fixes and other possible NULL dereferences [KC-23,KC-235]
- Memory leak fixes [KC-93,KC-98,KC-316]
- Reduction of symbol table sizes [KC-20]

12.17 Kopano Core 8.1.0

Kopano Core 8.1.0 is the first major release that went through a massive amount of quality assurance and therefore can be rated from us as a production-capable release. Kopano Core 8.1.0 is the first supported version from Kopano and provides you the following changes:

Fixes:

- ldapplugin: hopefully avoid “Timed out” errors from ldap_search [KC-74]
- swig: resolve a memory leak when using python components [KC-72]
- server: better guard against off-size EntryIDs [KC-60]
- ics: avoid referencing a value-replaced mysql_fetch_lengths array [KC-52]
- backup: restore container classes [KC-22]
- all daemons: fixed coredumps not getting generated most of the time [KC-61]
- all: issue setgroups(2) before setuid(2) [KC-37]

Enhancements:

- PHP7 support
- server: add ICS log messages [KC-18]
- server: ship example config files for ldap multi-server [KC-65]
- server: speed up REPLACE-type sql queries [KC-58]
- server: speed up login phase by caching PR_LOGON_TIME [KC-6]
- libicalmapi: improve fallback scenario to server_timezone in ical.cfg [KC-11]

Changes:

- search: no longer do indexing on root and Draft folders [KC-57]
- search: index junk folders, but skip updating suggestion list [KC-57]
- Python 2.5 support is removed

12.18 Kopano Core 8.0.1

Kopano Core 8.0.1 is the first major release of Kopano Core based on the open source code of ZCP (Zarafa Collaboration Platform). It marks the first iteration and we do not recommend this release yet as production release, even though many changes have been made.

A short list of changes are:

- server: avoid “netlink: 4 bytes leftover” in dmesg
- server: A fast-growing memory leak was resolved [KC-12]
- sql schema: all PRIMARY keys need to be NOT NULL, otherwise a UNIQUE key would be required. [KC-2]

Enhancements: - server: add ICS log messages [KC-18] - libicalmapi: improve fallback scenario to server_timezone in ical.cfg [KC-11]

Compiling from source

13.1 Installing Kopano Core from Source

KC is not officially supported by Kopano when build from source, yet in some situations - i.e. using KC on unsupported environments, or when preparing patches for KC - it is very useful to install from source. Since KC is distributed under an open source license (AGPLv3), it is in one's right to build KC from source.

How to exactly install KC from source is a procedure that is slightly different for each distribution and subject to change.

13.1.1 Requirements

The latest build and run time requirements can be obtained from <https://stash.kopano.io/projects/KC/repos/kopanocore/browse/doc/install.txt>.

Assuming the dependencies correctly installed, the basic build process is started with:

```
./configure --enable-epoll \  
            --enable-unicode \  
            --enable-python \  
            --disable-static \  
            --with-userscript-prefix=/etc/kopano/userscripts \  
            --with-quotatemplate-prefix=/etc/kopano/quotamail  
make  
make install
```

Important: Please note that builds from source are not covered by the support subscription. For support, please use the released builds which are quality tested by Kopano.

13.2 Installing Kopano MMC Snap-in from Source

To compile and install the Kopano AD Snap-in from source the following required tools need to be installed. Afterwards we are going to build the source and register the binaries to get the Kopano MMC Snap-in extension.

Once the binaries (DLLs) are built, these can be used on every other Windows box. *gacutil* is not included on a default Windows system, but it is possible to just copy *gacutil* and use it to register the binaries.

13.2.1 Requirements

- Windows system
- Remote Server Administration Tools for Windows
- 7-Zip (or any program to unzip the tar archive)
- MSBuild.exe (Microsoft Build Tools 2015)
- NuGet.exe
- RegAsm.exe (.NET Framework 4)
- gacutil.exe (.NET Framework 4.6 Software Development Kit)

13.2.2 Download the source code

The source code can be checked out from the [Kopano Git](#). The code for the extension can be found in the *mmc-plugin* subfolder.

13.2.3 Download and install Microsoft Build Tools 2015

Download and install Microsoft Build Tools 2015 for building the binaries:

```
PowerShell -Command "& {Invoke-WebRequest -Uri https://download.microsoft.com/
↳download/E/E/D/EEDF18A8-4AED-4CE0-BEBE-70A83094FC5A/BuildTools_Full.exe -OutFile
↳$env:TMP\BuildTools_Full.exe}
"%TMP%\BuildTools_Full.exe" /Quiet /NoRestart
```

13.2.4 Download NuGet

Download NuGet for downloading the dependencies:

```
PowerShell -Command "& {Start-BitsTransfer -Source https://dist.nuget.org/win-x86-
↳commandline/latest/nuget.exe -Destination $env:TMP}"
```

13.2.5 Download and install .NET Framework 4.6 Software Development Kit

Download and install NET Framework 4.6 Software Development Kit for registering the binaries:

```
PowerShell -Command "& {Invoke-WebRequest -Uri https://go.microsoft.com/fwlink/p/?
↳LinkID=822845 -OutFile $env:TMP\SDKSETUP.EXE}
"%TMP%\SDKSETUP.EXE" /features OptionId.NetFxSoftwareDevelopmentKit /quiet /
↳norestart
```

13.2.6 Build the binaries

Get the dependencies with NuGet and build the binaries with MSBuild:

```
nuget.exe restore "%TMP%\kopano_ad_extension_VERSION\mmc-
↳plugin\KopanoADS\KopanoADS.sln"
"%ProgramFiles(x86)%\MSBuild\14.0\Bin\MSBuild.exe" "%TMP%\kopano_ad_extension_
↳Version\mmc-plugin\KopanoADS\KopanoADS.sln"
```

13.2.7 Register the binaries

Register the binaries with RegAsm and gacutil, to do this a command line with administrator privileges is needed:

```
"%ProgramFiles(x86)%\Microsoft SDKs\Windows\v10.0A\bin\NETFX 4.6.2 Tools\gacutil.  
↵exe" -u Tulpep.ActiveDirectoryObjectPicker  
"%ProgramFiles(x86)%\Microsoft SDKs\Windows\v10.0A\bin\NETFX 4.6.2 Tools\gacutil.  
↵exe" -u KopanoADS  
"%SystemRoot%\Microsoft.NET\Framework64\v4.0.30319\RegAsm.exe" "%TMP%\kopano_ad_  
↵extension_VERSION\mmc-plugin\KopanoADS\Build\Debug\KopanoADS.dll"  
"%ProgramFiles(x86)%\Microsoft SDKs\Windows\v10.0A\bin\NETFX 4.6.2 Tools\gacutil.  
↵exe" -i "%TMP%\kopano_ad_extension_VERSION\mmc-  
↵plugin\KopanoADS\Build\Debug\Tulpep.ActiveDirectoryObjectPicker.dll"  
"%ProgramFiles(x86)%\Microsoft SDKs\Windows\v10.0A\bin\NETFX 4.6.2 Tools\gacutil.  
↵exe" -i "%TMP%\kopano_ad_extension_VERSION\mmc-  
↵plugin\KopanoADS\Build\Debug\KopanoADS.dll"
```

13.2.8 Verify installation

Verify if the MMC Snap-in extension installed correctly, there should be Kopano tabs available:

Eigenschaften von ? X

Organisation	Mitglied von	COM+	Kopano	Kopano Features
Allgemein	Adresse	Konto	Profil	Rufnummern



Vorname: Initialen:

Nachname:

Anzeigename:

Beschreibung:

Büro:

Rufnummer:

E-Mail:

Webseite:

14.1 Upgrade from Zarafa Collaboration Platform

Upgrading from Zarafa Collaboration Platform to Kopano is technically possible, yet only supported for upgrading from versions of ZCP 7.2. Older installations of ZCP should be upgraded to ZCP 7.2 first, before upgrading to Kopano 8.

Appendix B: LDAP attribute description

This appendix will describe all available LDAP attributes available in the Kopano schema. The Kopano schema is available in the directory `/usr/share/doc/kopano`.

Please keep in mind that the Kopano LDAP configuration files are very flexible, so these attributes are not in all cases used.

`kopanoQuotaOverride`

This attribute is used to override the default quota, which is configured in the `/etc/kopano/server.cfg`. This attribute always need to be enabled to use a custom quota setting.

OID	1.3.6.1.4.1.47732.1.1.1 .1
Syntax	Integer
Multi- or Single-Valued	Single-Valued

`kopanoQuotaWarn`

This attribute contains the warning quota level in Mb.

OID	1.3.6.1.4.1.47732.1.1.1 .2
Syntax	Integer
Multi- or Single-Valued	Single-Valued

`kopanoQuotaSoft`

This attribute contains the soft quota level in Mb.

OID	1.3.6.1.4.1.47732.1.1.1 .3
Syntax	Integer
Multi- or Single-Valued	Single-Valued

`kopanoQuotaHard`

This attribute contains the hard quota level in Mb.

OID	1.3.6.1.4.1.47732.1.1.1 .4
Syntax	Integer
Multi- or Single-Valued	Single-Valued

`kopanoUserDefaultQuotaOverride`

This attribute will override the system wide quota settings for all users of the company.

OID	1.3.6.1.4.1.47732.1.1.1 .5
Syntax	Integer
Multi- or Single-Valued	Single-Valued

`kopanoUserDefaultQuotaWarn`

This attribute contains the warning quota level in Mb for all users of the company.

OID	1.3.6.1.4.1.47732.1.1.1 .6
Syntax	Integer
Multi- or Single-Valued	Single-Valued

`kopanoUserDefaultQuotaSoft`

This attribute contains the soft quota level in Mb for all users of the company.

OID	1.3.6.1.4.1.47732.1.1.1 .7
Syntax	Integer
Multi- or Single-Valued	Single-Valued

`kopanoUserDefaultQuotaHard`

This attribute contains the hard quota level in Mb for all users of the company.

OID	1.3.6.1.4.1.47732.1.1.1 .8
Syntax	Integer
Multi- or Single-Valued	Single-Valued

`kopanoAdmin`

This attribute will make a user Kopano administrator.

OID	1.3.6.1.4.1.47732.1.1.2 .1
Syntax	Integer
Multi- or Single-Valued	Single-Valued

`kopanoSharedStoreOnly`

This attribute will configure a mailbox as a shared store. On shared stores you will not be able to login.

OID	1.3.6.1.4.1.47732.1.1.2 .2
Syntax	Integer
Multi- or Single-Valued	Single-Valued

`kopanoAccount`

This attribute can be used in the LDAP search filters to filter users and groups.

OID	1.3.6.1.4.1.47732.1.1.2 .3
Syntax	Integer
Multi- or Single-Valued	Single-Valued

`kopanoSendAsPrivilege`

This attribute will contain users or groups that should have sendas permissions on the user where this attribute is added.

OID	1.3.6.1.4.1.47732.1.1.2 .4
Syntax	DN or DirectoryString
Multi- or Single-Valued	Multi-Valued

`kopanoMrAccept`

This attribute will configure auto-acceptance of meeting requests. This attribute is `not` used in the current Kopano versions.

OID	1.3.6.1.4.1.47732.1.1.2 .5
Syntax	Integer
Multi- or Single-Valued	Single-Valued

`kopanoMrDeclineConflict`

This attribute will decline meeting requests when the calendar already contains appointments. This attribute is `not` used in the current Kopano versions.

OID	1.3.6.1.4.1.47732.1.1.2 .6
Syntax	Integer
Multi- or Single-Valued	Single-Valued

`kopanoMrDeclineRecurring`

This attribute will decline meeting requests when they are set as recurrent. This attribute is `not` used in the current Kopano versions.

OID	1.3.6.1.4.1.47732.1.1.2 .7
Syntax	Integer
Multi- or Single-Valued	Single-Valued

`kopanoId`

This attribute can be used as a generic unique id for example users and groups. This attribute is by default `not` used by Kopano, but can be used for example together with identity management solutions.

OID	1.3.6.1.4.1.47732.1.1.2 .8
Syntax	Integer
Multi- or Single-Valued	Single-Valued

`kopanoResourceType`

This attribute will configure the resource type of a shared store. The available options are `Room` or “Equipment”

OID	1.3.6.1.4.1.47732.1.1.2 .9
Syntax	DirectoryString
Multi- or Single-Valued	Single-Valued

`kopanoResourceCapacity`

This attribute will number the rooms or equipment available.

OID	1.3.6.1.4.1.47732.1.1.2 .10
Syntax	Integer
Multi- or Single-Valued	Single-Valued

`kopanoHidden`

This attribute will hide the object in the Global Address Book. This will also hide the object for administrator users.

OID	1.3.6.1.4.1.47732.1.1.2 .11
Syntax	Integer
Multi- or Single-Valued	Single-Valued

`kopanoEnabledFeatures`

Controls which features are explicitly enabled for a user, and overrides any disabled features in the server `disabled_features` setting.

OID	1.3.6.1.4.1.47732.1.1.2 .13
Syntax	String
Multi- or Single-Valued	Multi-Valued

`kopanoDisabledFeatures`

Controls which features are explicitly disabled for a user.

OID	1.3.6.1.4.1.47732.1.1.2 .14
Syntax	String
Multi- or Single-Valued	Multi-Valued

`kopanoAliases`

This attribute will contain all other email addresses and aliases for the user.

OID	1.3.6.1.4.1.47732.1.1.3 .1
Syntax	DirectoryString
Multi- or Single-Valued	Multi-Valued

`kopanoUserServer`

This attribute will be the homeserver of a user when running in multi-server mode.

OID	1.3.6.1.4.1.47732.1.1.4 .1
Syntax	DirectoryString
Multi- or Single-Valued	Single-Valued

`kopanoSecurityGroup`

This attribute will specify whether a group has security privileges. When the attribute is set to 0, the group will be seen as distribution list.

OID	1.3.6.1.4.1.47732.1.2.2 .1
Syntax	Integer
Multi- or Single-Valued	Single-Valued

`kopanoViewPrivilege`

This attribute will contain companies with view privileges over the selected company.

OID	1.3.6.1.4.1.47732.1.3.2 .4
Syntax	DirectoryString
Multi- or Single-Valued	Multi-Valued

`kopanoAdminPrivilege`

This attribute will contain users from different companies which are administrator over selected company.

OID	1.3.6.1.4.1.47732.1.3.2 .5
Syntax	DirectoryString
Multi- or Single-Valued	Multi-Valued

`kopanoSystemAdmin`

This attribute will specify the users who are system administrators for this company.

OID	1.3.6.1.4.1.47732.1.3.2 .6
Syntax	DirectoryString
Multi- or Single-Valued	Multi-Valued

`kopanoQuotaUserWarningRecipients`

This attribute will contain users who will receive a notification email when a user exceeds his quota.

OID	1.3.6.1.4.1.47732.1.3.1 .5
Syntax	DirectoryString
Multi- or Single-Valued	Multi-Valued

`kopanoQuotaCompanyWarningRecipients`

This attribute will contain email address who will receive a notification email when a company exceeds his quota.

OID	1.3.6.1.4.1.47732.1.3.1 .6
Syntax	DirectoryString
Multi- or Single-Valued	Multi-Valued

`kopanoCompanyServer`

This attribute will contain the home server of a company when running in multi-server mode.

OID	1.3.6.1.4.1.47732.1.3.4 .1
Syntax	DirectoryString
Multi- or Single-Valued	Single-Valued

`kopanoHttpPort`

This attribute will contain the port for the http connections when running in multi-server mode.

OID	1.3.6.1.4.1.47732.1.4.4 .1
Syntax	Integer
Multi- or Single-Valued	Single-Valued

`kopanoSslPort`

This attribute will contain the port for the https connections when running in multi-server mode.

OID	1.3.6.1.4.1.47732.1.4.4 .2
Syntax	Integer
Multi- or Single-Valued	Single-Valued

`kopanoFilePath`

This attribute will contain the unix socket or the named pipe of the server when running in multi-server mode.

OID	1.3.6.1.4.1.47732.1.4.4 .3
Syntax	DirectoryString
Multi- or Single-Valued	Single-Valued

`kopanoContainsPublic`

This attribute will enable the public store for a specific multi-server node. Make sure only one node has enabled this attribute.

OID	1.3.6.1.4.1.47732.1.4.4 .4
Syntax	Integer
Multi- or Single-Valued	Single-Valued

`kopanoFilter`

This attribute will contain the LDAP filter to apply for an addresslist or dynamic group.

OID	1.3.6.1.4.1.47732.1.5.5 .1
Syntax	DirectoryString
Multi- or Single-Valued	Single-Valued

`kopanoBase`

This attribute will contain the LDAP search base to apply for an addresslist or dynamic group.

OID	1.3.6.1.4.1.47732.1.5.5 .2
Syntax	DirectoryString
Multi- or Single-Valued	Single-Valued

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: kopano
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: kopano-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: kopano-user
objectClass: inetOrgPerson
gidNumber: 1000
cn: John Doe
homeDirectory: /home/john
mail: john@example.com
uidNumber: 1000
kopanoAliases: j.doe@example.com
kopanoUserServer: node1
uid: john
kopanoAccount: 1
kopanoAdmin: 0
sn: Doe
userPassword: john
kopanoQuotaOverride: 1
kopanoEnabledFeatures: imap
kopanoDisabledFeatures: pop3
kopanoQuotaWarn: 1000000000
kopanoQuotaSoft: 1100000000
kopanoQuotaHard: 1200000000

dn: cn=Example addresslist,ou=Addresslists,dc=example,dc=com
objectClass: kopano-addresslist
objectClass: top
cn: Example addresslist
kopanoFilter: (mail=*@example.com)

dn: cn=Example security group,ou=Groups,dc=example,dc=com
objectClass: posixGroup
objectClass: top
objectClass: kopano-group
kopanoHidden: 0
cn: Example security group
gidNumber: 1000
memberUid: john
kopanoAccount: 1
description: Example security group
kopanoSecurityGroup: 1

dn: cn=Example distribution group,ou=Groups,dc=example,dc=com
objectClass: posixGroup
objectClass: top
objectClass: kopano-group
kopanoHidden: 0
cn: Example distribution group
memberUid: john
kopanoAccount: 1
gidNumber: 1001
description: Example distribution group
kopanoSecurityGroup: 0

```

Appendix D: Common MAPI Errors

This Table shows the most common MAPI error codes and their corresponding MAPI error name which allow better identification of the reason why a MAPI transaction has failed:

Error Code	Error Name
0x80004002	MAPI_E_INTERFACE_NOT_SUPPORTED
0x80004005	MAPI_E_CALL_FAILED
0x80070005	MAPI_E_NO_ACCESS
0x8007000e	MAPI_E_NOT_ENOUGH_MEMORY
0x80070057	MAPI_E_INVALID_PARAMETER
0x80040102	MAPI_E_NO_SUPPORT
0x80040103	MAPI_E_BAD_CHARWIDTH
0x80040105	MAPI_E_STRING_TOO_LONG
0x80040106	MAPI_E_UNKNOWN_FLAGS
0x80040107	MAPI_E_INVALID_ENTRYID
0x80040108	MAPI_E_INVALID_OBJECT
0x80040109	MAPI_E_OBJECT_CHANGED
0x8004010a	MAPI_E_OBJECT_DELETED
0x8004010b	MAPI_E_BUSY
0x8004010d	MAPI_E_NOT_ENOUGH_DISK
0x8004010e	MAPI_E_NOT_ENOUGH_RESOURCES
0x8004010f	MAPI_E_NOT_FOUND
0x80040110	MAPI_E_VERSION
0x80040111	MAPI_E_LOGON_FAILED
0x80040112	MAPI_E_SESSION_LIMIT
0x80040113	MAPI_E_USER_CANCEL
0x80040114	MAPI_E_UNABLE_TO_ABORT
0x80040115	MAPI_E_NETWORK_ERROR
0x80040116	MAPI_E_DISK_ERROR
0x80040117	MAPI_E_TOO_COMPLEX
0x80040118	MAPI_E_BAD_COLUMN
0x80040119	MAPI_E_EXTENDED_ERROR
0x8004011a	MAPI_E_COMPUTED
0x8004011b	MAPI_E_CORRUPT_DATA
0x8004011c	MAPI_E_UNCONFIGURED

Continued on next page

Table 17.1 – continued from previous page

Error Code	Error Name
0x8004011d	MAPI_E_FAILONEPROVIDER
0x8004011e	MAPI_E_UNKNOWN_CPID
0x8004011f	MAPI_E_UNKNOWN_LCID
0x80040120	MAPI_E_PASSWORD_CHANGE_REQUIRED
0x80040121	MAPI_E_PASSWORD_EXPIRED
0x80040122	MAPI_E_INVALID_WORKSTATION_ACCOUNT
0x80040123	MAPI_E_INVALID_ACCESS_TIME
0x80040124	MAPI_E_ACCOUNT_DISABLED
0x80040200	MAPI_E_END_OF_SESSION
0x80040201	MAPI_E_UNKNOWN_ENTRYID
0x80040202	MAPI_E_MISSING_REQUIRED_COLUMN
0x00040203	MAPI_W_NO_SERVICE
0x80040301	MAPI_E_BAD_VALUE
0x80040302	MAPI_E_INVALID_TYPE
0x80040303	MAPI_E_TYPE_NO_SUPPORT
0x80040304	MAPI_E_UNEXPECTED_TYPE
0x80040305	MAPI_E_TOO_BIG
0x80040306	MAPI_E_DECLINE_COPY
0x80040307	MAPI_E_UNEXPECTED_ID
0x00040380	MAPI_W_ERRORS_RETURNED
0x80040400	MAPI_E_UNABLE_TO_COMPLETE
0x80040401	MAPI_E_TIMEOUT
0x80040402	MAPI_E_TABLE_EMPTY
0x80040403	MAPI_E_TABLE_TOO_BIG
0x80040405	MAPI_E_INVALID_BOOKMARK
0x00040481	MAPI_W_POSITION_CHANGED
0x00040482	MAPI_W_APPROX_COUNT
0x80040500	MAPI_E_WAIT
0x80040501	MAPI_E_CANCEL
0x80040502	MAPI_E_NOT_ME
0x00040580	MAPI_W_CANCEL_MESSAGE
0x80040600	MAPI_E_CORRUPT_STORE
0x80040601	MAPI_E_NOT_IN_QUEUE
0x80040602	MAPI_E_NO_SUPPRESS
0x80040604	MAPI_E_COLLISION
0x80040605	MAPI_E_NOT_INITIALIZED
0x80040606	MAPI_E_NON_STANDARD
0x80040607	MAPI_E_NO_RECIPIENTS
0x80040608	MAPI_E_SUBMITTED
0x80040609	MAPI_E_HAS_FOLDERS
0x8004060a	MAPI_E_HAS_MESSAGES
0x8004060b	MAPI_E_FOLDER_CYCLE
0x8004060c	MAPI_E_STORE_FULL
0x8004060D	MAPI_E_LOCKID_LIMIT
0x00040680	MAPI_W_PARTIAL_COMPLETION
0x80040700	MAPI_E_AMBIGUOUS_RECIP
0x80040800	SYNC_E_OBJECT_DELETED
0x80040801	SYNC_E_IGNORE
0x80040802	SYNC_E_CONFLICT
0x80040803	SYNC_E_NO_PARENT
0x80040804	SYNC_E_INCEST
0x80040805	SYNC_E_UNSYNCHRONIZED
0x00040820	SYNC_W_PROGRESS

Continued on next page

Table 17.1 – continued from previous page

Error Code	Error Name
0x00040821	SYNC_W_CLIENT_CHANGE_NEWER

Copyright © 2016 Kopano

Adobe, Acrobat, Acrobat Reader and PostScript are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Apache is a trademark of The Apache Software Foundation.

Apple, Mac, Macintosh, Mac OS, iOS, Safari and TrueType are trademarks of Apple Computer, Inc., registered in the United States and other countries.

Blackberry is the trademark or registered trademark of BlackBerry Limited, the exclusive rights to which are expressly reserved. Kopano is not affiliated with, endorsed, sponsored, or otherwise authorized by BlackBerry Limited.

Collax is a trademark of Collax GmbH.

Debian is a registered trademark of Software in the Public Interest, Inc.

ECMAScript is the registered trademark of Ecma International.

Gentoo is a trademark of Gentoo Foundation, Inc.

Google, Android and Google Chrome are trademarks or registered trademarks of Google Inc.

IBM and PowerPC are trademarks of International Business Machines Corporation in the United States, other countries, or both.

MariaDB is a registered trademark of MariaDB Corporation AB.

Microsoft, Microsoft Internet Explorer, the Microsoft logo, the Microsoft Internet Explorer logo, Windows, Windows Phone, Office Outlook, Office 365, Exchange, Active Directory and the Microsoft Internet Explorer interfaces are trademarks or registered trademarks of Microsoft, Inc.

Mozilla, Firefox, Mozilla Firefox, the Mozilla logo, the Mozilla Firefox logo, and the Mozilla Firefox interfaces are trademarks or registered trademarks of Mozilla Corporation.

MySQL, InnoDB, JavaScript and Oracle are registered trademarks of Oracle Corporation Inc.

NDS and eDirectory are registered trademarks of Novell, Inc.

NGINX is a registered trademark of Nginx Inc. NGINX Plus is a registered trademark of Nginx Inc.

Opera and the Opera “O” are registered trademarks or trademarks of Opera Software AS in Norway, the European Union and other countries.

Postfix is a registered trademark of Wietse Zweitze Venema.

QMAIL is a trademark of Tencent Holdings Limited.

Red Hat, Red Hat Enterprise Linux, Fedora, RHCE and the Fedora Infinity Design logo are trademarks or registered trademarks of Red Hat, Inc. in the U.S. and other countries.

SUSE, SLES, SUSE Linux Enterprise Server, openSUSE, YaST and AppArmor are registered trademarks of SUSE LLC.

Sendmail is a trademark of Sendmail, Inc.

UNIX is a registered trademark of The Open Group.

Ubuntu and Canonical are registered trademarks of Canonical Ltd.

Univention is a trademark of Ganten Investitions GmbH.

All trademarks are property of their respective owners. Other product or company names mentioned may be trademarks or trade names of their respective owner.

Disclaimer: Although all documentation is written and compiled with care, Kopano is not responsible for direct actions or consequences derived from using this documentation, including unclear instructions or missing information not contained in these documents.

The text of and illustrations in this document are licensed by Kopano under a Creative Commons Attribution–Share Alike 3.0 Unported license (“CC-BY-SA”). An explanation of CC-BY-SA is available at [the creativecommons.org website](http://creativecommons.org). In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version. This document uses parts from the Zarafa Collaboration Platform (ZCP) Administrator Manual, previously located at https://documentation.zarafa.com/zarafa_changelog, licensed under CC-BY-SA.