Kopano Archiver Manual

Release 8.2.0

Kopano BV

Oct 23, 2018

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

CHAPTER 1

Introduction

The Kopano Archiver provides a Hierarchical Storage Management (HSM) solution for Kopano, with the following features and advantages:

- More rapid recovery from downtime incidents
- Improved messaging server performance
- Seamless End User Experience
- Flexible PST Migration
- Use less expensive storage tiers for long-term archiving

With the Kopano Archiver older messages will be automatically moved to slower and thus cheaper storage. The slow storage consists of one or more additional Kopano Archive servers which sole task it is to store archived messages.

To activate the archiving function for users, they need to be coupled to an archive mailbox. The archive mailboxes will be located on one of the archive servers. The archive servers have exactly the same storage architecture as a normal Kopano server, all MAPI properties are stored in a MySQL database and all attachments are stored compressed on disk.

Although the older messages are stored on different servers, the user will not notice this as the Kopano Archiver provides transparent stubbing. Older messages can still be accessed from the main mailbox of the user, by using the (de)stubbing feature. Once a message is archived, it may be deleted from the original store. Kopano Archiver can create a stub to the archived message, allowing a user to view the archived message and open it as if it were a normal message.

The Kopano Archiver uses the Kopano's multi-server technology to access archive stores in a seamless way using transparent user authentication. Nonetheless, the Kopano Archiver can be used in a single-server setup with limited functionality.

Kopano Archiver is an additional product to Kopano Core and is not provided as a default component.

CHAPTER 2

Conventions

Please read this chapter before installing and deploying Kopano Archiver, in order to understand its terminology.

- **Primary Server**

The primary server is the server with the best performance and best I/O subsystem, that contains the mailboxes of the users holding the most recent data.

Note: Although the term Primary Server suggests that there's only one primary server, multiple primary servers can exist in a multi-server environment. In this document no distinction will be made between a single-server or multi-server environment unless explicitly stated.

- **Archive Server**

An archive server is the server that contains the archives for stores that reside on the primary server. An archive server is a Kopano server with the sole purpose of providing storage for one or more archive stores. In a multi-server environment this server will be just another node in the cluster. Unlike primary servers, there's no need for a multi-server environment to have a multi-archive server setup.

Note: An archive server is another kopano-server with the sole purpose of providing storage for one or more archive stores. In a multi-server environment this server will be just another node in the cluster.

Note: Unlike primary servers, there's no need for a multi-server environment to have multiple archive servers.

- **Primary Store**

The primary store is the store that resides on a primary server and on which a user normally works. Also known as **Main Store**

- **Archive Store**

The archive store is the store that resides on an archive server and which is used for storing archived messages from the primary store.

- **kopano-archiver**

The archiver is the application to manage the archiving. Basically it can be used to attach primary stores to archive stores and execute archive runs. It can be installed on any Kopano server to connect to the primary or archive server using SSL authentication. It can also be used on a single server, using **kopano-server**'s unix socket.

- **Stubbed Message**

A stubbed message is a message in the primary store that acts as a placeholder for the archived message. These messages occupy virtually no space in the primary store, but show a user that a message was once there. It also acts as an entry point to the archived copy of that message.

- **Single Instances**

Kopano uses single instance storage whenever possible in order to minimise storage requirements when data is stored more than once. Kopano Archiver makes use of this technology by remembering which instances it copied to an archive server and referencing that instance whenever possible.

- **Attached/detached stores**

An archive store is attached, when a data link is made with a primary store. Making such a link is called *attaching* a store, removing such a link *detaching* it. Detached stores are still available for reading.

- **Hooked/unhooked stores**

A hooked store is available for use as a primary or archive store. Unhooked stores are not available. From the perspective of a mail client or the Archiver they don't exist. Such unhooked stores are also know as **orphaned stores**. Unhooked stores cannot be read from or written to. The process of making a store available for a user is called *hooking*, the reverse *unhooking*. An unhooked store may be removed.

- **Archive Configuration**

An archive store can be configured in two ways, one-for-one and one-for-many. This is not a system wide configuration and can be setup for each archive independently. This allows for hybrid systems where N users with small to medium stores can be placed on M archive stores (where M is significantly smaller than N) and users with big to huge stores can be placed on dedicated archive stores.

- **One-for-One Configuration**

In a one-for-one configuration one archive store is attached to one primary store. The advantage of this configuration is that it's faster as the archive store itself is kept smaller. The disadvantage is that for each user an additional non-active user needs to be created (since there's a one-to-one mapping between stores and users in Kopano).

- **One-for-Many Configuration**

In a one-for-many configuration one archive store is attached to multiple primary stores. For each attached primary store a folder is created in the archive store that will act as the root of the archive for that particular primary store. The advantage of this configuration is that less additional non-active users are required. The disadvantage is that the archive will become slower if the total amount of archived data in it grows.

CHAPTER 3

Hardware considerations

Sizing and implementing Kopano Archiver requires careful planning to ensure that the product can perform to expectations and scaling as the customer requirements grow and fits in the existing infrastructure.

3.1 CPU

In most deployment scenarios the Kopano Archiver will be installed on a dedicated server which doesn't provide any other services. The most heavy actions performed on the archive server is the actual archive run, which is typically done once or twice a day. This archive run will in general do a lot of database transactions and is not running CPU intensive calculations.

The archiver run will archive the mailboxes one by one. For the right choice of the amount of CPU's it's recommended to use a multi-core system where one core is reserved for the Kopano-server, one for the MySQL database server and one for the actual Archive controller.

The advised amount of CPU cores is 4 where the prefered archicture is 64bits, so more than 4Gb of memory can be allocated by processes.

3.2 Memory

In normal setups of Kopano Core, memory is one of the most important hardware components as both MySQL and Kopano-server cache requested data, so the second time the calendar or inbox is opened all items will be retrieved from the cache. In an archiving setup with the stubbing feature enabled, the archived data is only accessed when an archived email is really opened. In this case the advantage of caching is limited, as the archived emails are normally not accessed over and over again. However when the users open the archive store as a secondary mailbox, caching is more important as the user will directly access the archive store.

The advised amount of memory is:

Amount of archive users	Stubbing enabled	No stubbing
<500	2GB	4GB
500 - 2000	4GB	8GB
>2000	8GB	16GB

3.3 Storage

In most cases, you require RAID-based storage to achieve your storage requirements. To maintain performance and reliability, consider hardware-based RAID rather than software-based RAID. To achieve redundancy on striped arrays while maintaining performance, consider the RAID scheme carefully.

RAID level 5 is popular to arrange cost-effective methods of achieving redundancy while keeping a good read performance. However, write actions on RAID level 5 will cost performance for storing parity bits. Therefore, in most cases discussed, a RAID1 or RAID10 should be considered. The RAID controller should also provide a battery-backed read and write cache to aid performance and prevent data corruption at power failures.

Before you use partitions on a storage area network (SAN), consider the I/O load together with any other applications that are already using the SAN to ensure that the performance can be maintained.

Ideally, discuss the implementation with the storage vendor or responsible to ensure that you achieve the best performance. Typically, you should create LUNs across as many suitable disks as possible, using entire disks rather than partial disks to prevent multiple I/O-intensive applications from using the same disks.

3.4 Virtual vs Physical

When above storage and memory requirements are matched, the Kopano Archiver can run on both virtual as physical servers. Most organisation use the Kopano Archiver on a virtual platform, as this is the default in todays data center environments.

CHAPTER 4

Installation

4.1 Requirements

To deploy the Kopano Archiver at least two servers are required: one primary Kopano Core server and one Kopano Archive server. Multi-server technology is used to connect archives to users automatically.

Note: A multi-server setup requires a central LDAP or Active Directory. It's not possible to use multi-server with the `DB` or `unix` user plugin.

4.1.1 Software Requirements

- Kopano Core 8+
- Multi-server Kopano setup with LDAP or ADS (please refer to the Kopano Core Administrator manual for more information on how to setup a Multi-Server Environment which you can find here:)
- Kopano subscription including Archiver component

4.1.2 Required Packages

The following Kopano packages are required on a primary node:

- kopano-client
- kopano-dagent (for LMTP only)
- kopano-search or kopano-indexer
- kopano-server
- kopano-spooler
- kopano-utils

The following Kopano packages are required on an archive node:

- python-mapi
- kopano-client
- kopano-common
- kopano-libarchiver
- kopano-libs
- kopano-server

- kopano-utils

4.2 Installation

The Kopano Archiver software can be found on the Kopano Portal as an additional download or installed via the Kopano repository archiver: The Kopano Archiver download will include the following package:

- kopano-archiver containing the Archive controller and configuration files

4.2.1 Default installation location

The Kopano Archiver packages can be installed on any node in a multi-server setup. However, installing the packages on the archive node is recommended.

RPM based distributions

Use the following command to install the `kopano-archiver` package on RPM based distributions:

```
rpm -Uvh kopano-archiver_<version>_<platform>.rpm
```

Replace `<version>` with the correct version and `<platform>` with the required target platform (`i386`, `i586`, `x86_64`).

DEB based distributions

On Debian based distributions use:

```
dpkg -i kopano-archiver_<version>_<platform>.deb
```

Replace `<version>` with the correct version and `<platform>` with the required target platform (`i386`, `x86_64`).

4.3 Basic Configuration

Kopano Archiver is configured by default in `/etc/kopano/archiver.cfg`. If the default configuration file is not found, Archiver will try to work with default configuration settings, which will most probably fail. Archiver will display an error message like “Unable to open admin session on server *https://192.168.1.10:237/kopano*” and close.

An alternative configuration file may be specified by using command line option `--config`. When such a configuration file is specified and cannot be found, Archiver will emit an error message like: “Unable to open configuration file /tmp/test.cfg” and close.

4.3.1 Connection Parameters

As with all Kopano components, `kopano-archiver` needs to know where to connect to and how to authenticate. This is configured using the `server_socket`, `sslkey_file` and `sslkey_pass` settings.

For instance:

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

Or when running the archiver an a different server:

```
server_socket = https://servername:237/kopano
sslkey_file = /etc/kopano/ssl/client.pem
sslkey_pass = secret
```

Note: Under normal circumstances `kopano-archiver` will connect to a local server through the file socket.

4.3.2 MySQL Settings

`kopano-archiver` uses one central MySQL database for managing deduplication of archived attachments. MySQL settings can be configured like this:

```
mysql_host = localhost
mysql_port = 3306
mysql_user = kopano
mysql_password = password
mysql_socket =
mysql_database = kopano_archiver
```

This database should not be the same one that `kopano-server` is using and needs to be accessible over the network from each server running the `kopano-archiver` command.

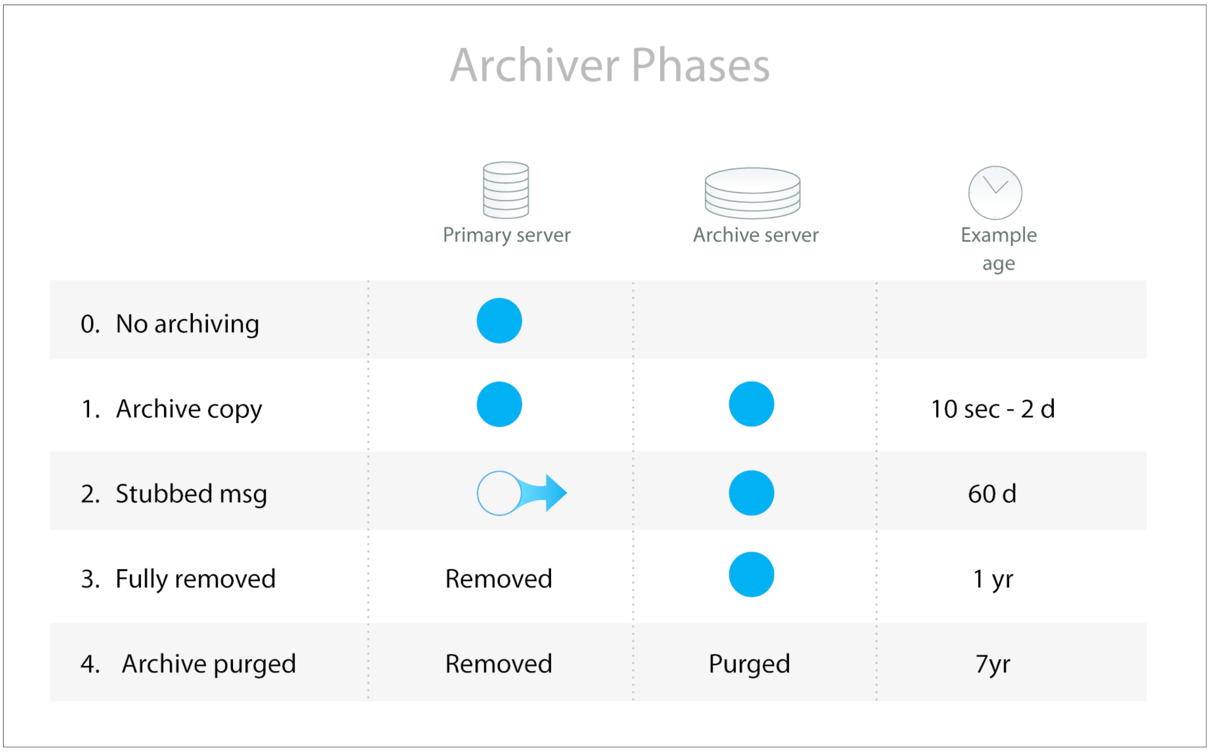
CHAPTER 5

Archive configuration

`kopano-archiver` is configured in a configuration file, that can be specified on the command line using command line option `--config` or `-c`. When no configuration file is specified from the command line, Archiver tries to find a default configuration file named `archiver.cfg` located in `/etc/kopano/`. Please note that when no configuration file can be found, information required to proceed is missing. In such a case Archiver will emit an error message and exit.

Depending on user requirements, Archiver can be configured in many different ways. Five different stages in Archiving can be distinguished, of which any subset can be used. This chapter describes a basic setup. Later more advanced configurations using more complex combinations of stages will be presented.

The five stages of Archiving are:



Archiver Phases

	Primary server	Archive server	Example age
0. No archiving	●		
1. Archive copy	●	●	10 sec - 2 d
2. Stubbed msg	○➜	●	60 d
3. Fully removed	Removed	●	1 yr
4. Archive purged	Removed	Purged	7yr

Figure 4.1. Archiver stages

Archiving solutions have different stages to process the email data. Below the four stages of the archive process are described. The ages listed in the rightmost colums are examples only.

Every of these stages can be configured in the configuration file of the Archive Controller. Before deploying the Kopano Archiver make sure the configuration options are carefully set according to the archive strategy. The default installed configuration will only copy message items from linkes mailboxes to the archive mailboxes.

Table 4.1. Archiver stages

Stage	Description
No archiving	No archive store exists and no messages are archived.
Archive copy	All messages that are eligible for archiving are copied from the primary store to the archive store. When a copy of a message is made, an internal reference to this copy is placed in the original message.
Stubbed message	All messages that are eligible for stubbing are stubbed. A stub is defined as the original message with the body and attachments removed. A message is eligible for stubbing when it reaches the specified minimum age AND archived copies are present. So a message is never stubbed if it's not yet copied to the archive store.
Fully removed (Delete)	All messages and stubs that are eligible for deletion are removed from the primary store. A message is eligible for deletion when it reaches the specified minimum age AND archived copies are present. So a message is never deleted from the primary store if it's not yet copied to the archive store.
Archive purged	Messages that reached a specific age will be deleted completely from the archive store. In this case messages will not be available on both the primary and archive server.

5.1 Scenarios

Email archiving are typically designed for two different scenarios.

- Email archiving for storage purposes
- Email archiving for keeping an email history and compliancy

Kopano Archiver can be configured for both purposes.

5.1.1 Email archiving for storage purposes

The volume of incoming emails is still growing at most organisations, where most mailbox quotas will not be increased. Users will organise and archive emails to pst files to keep their mailbox clean.

By implementing the Kopano Archiver all older emails can be stored on central located archive server. Where the archive database is typically stored on slower and cheaper storage. Storing the older emails on a different server will result in a smaller mailboxes of the primary server and therefore a better performant primary server. By having the archived emails stored centrally, the disadvantages of decentrally stored email archives (such as pst files) files can be eliminated:

- Local archives can not be accessed by multiple users
- Local archives, especially PST files are unmanageable by organisation (e.g. not part of backup)
- PST files have slow network performance

With the stubbing feature the user can transparently access emails from the primary mailbox, where the actual body and attachment of the emails are accessed from the archive server. With the direct connection to the archive mailboxes, users can access there archived emails which are not available as stub in the primary mailbox.

5.1.2 Email archiving for keeping an email history

As email is used for more and more official communication, companies want to ensure no emails are lost. Email archiving solutions for keeping email histories will automatically archive all incoming and outgoing emails. All archived emails will be stored on a central archive server and can be kept for long-term accountability.

As normally all users access to their archive mailbox, so users can still access older emails.

With the Kopano Archiver all incoming and outgoing emails can be directly archived after the send/delivery process.

The Kopano Archiver controller can still archive, stub, and cleanup primary mailboxes to reduce the size of mailboxes, however in this case the emails that were already in the archive database will not be modified.

5.2 Basic Archiver configuration

In a setup all stages as shown in Figure 4.1 can be configures. It should be noted, though, that not all combinations make sense. As a basic setup for Kopano Archiver, considered best practice, is a `stubbing-only` configuration. In this setup all emails can be accessed from the primary store of the user. Depending on the age of the message the email will be opened on the primary server or a stub will open the item directly in the user's archive mailbox. In this case the user doesn't directly access his or her archive mailbox to view emails.

For a `stubbing only` setup settings like the following must be configured in `/etc/kopano/archiver.cfg`:

```
archive_enable = yes
archive_after = 90

stub_enable = yes
stub_after = 90
stub_unread = no

delete_enable = no
delete_after = 0

purge_enable = no
purge_after = 0

cleanup_action = store
cleanup_follow_purge_after = no

enable_auto_attach = yes
auto_attach_writable = no
```

Setting `archive_enable = yes` enables archive operation, i.e. copying emails from the primary node to the archive node. Setting `archive_after = 90` indicates that messages older than 90 days are to be archived.

Setting `stub_enable = yes` enables stubbing. Body and attachments messages in the primary store are removed and a reference to the archive store is created. Setting `stub_after = 90` means that messages older than 90 days will be stubbed.

Unread emails are not stubbed by default. Setting `stub_unread = yes` means that even unread messages are stubbed.

Settings `delete_enable` and `purge_enable` must be set to `no`, since a user will not access his archive mailbox directly in this mode.

Setting `cleanup_action = store` means that, in a cleanup run, archived messages of which stubs have been deleted from the archive mailbox will be moved to an archive folder named `Kopano Archive\Deleted`. See *Cleanup* for an explanation of this cleanup process.

Setting `enable_auto_attach = yes` means that a user automatically gets an archive mailbox when the archive server option is enabled in Active Directory or OpenLDAP. The user will always get read-only permissions, so users can delete messages from the primary mailbox only. This is required to make sure the stubs in the primary mailbox will always point to a existing message.

5.3 Archive-only setup

Another setup is archive-only, in which emails will exist either in the primary mailbox or in an archive mailbox. Stubbing is disabled in this mode.

For an archive-only setup the following settings need to be configured in `/etc/kopano/archiver.cfg`:

```
archive_enable = yes
archive_after = 365
```

```
stub_enable = no

delete_enable = yes
delete_after = 365

purge_enable = yes
purge_after = 3650

enable_auto_attach = yes
auto_attach_writable = yes
```

To move items to the archive mailbox setting delete_enable is set to yes and the number of days is configured in delete_after. In this example, items older than 1 year are deleted from the primary mailbox and after that can only be accessed from the archive mailbox.

Setting stub_enable = no makes sure there will be no stubs. Since there are no stubs, there is no sense in doing cleanup runs, so settings for cleanup need not be given.

To clean up the archive mailbox, items will be completely removed after 10 years.

As stubbing is not used in this mode, the archives can be set writable so users can clean up their archive mailbox themselves.

5.4 Advanced setups

Advanced setups are possible, but may cause some confusion initially, because some settings interfere. An example of an advanced setup is the following configuration, that resembles the setup from Figure 4.1:

```
archive_enable = yes
archive_after = 2

stub_enable = yes
stub_after = 60
stub_unread = no

delete_enable = yes
delete_after = 365
stub_unread = yes

purge_enable = yes
purge_after = 3650

cleanup_action = store
cleanup_follow_purge_after = yes

enable_auto_attach = yes
auto_attach_writable = no
```

In this example mode emails will be archived after two days as an archived copy, while the original remains in main store. After 60 days the copy from main store will be removed and replaced by a stub to the archived copy. After a year, the stub will be removed, while the copy in the archive remains.

To prevent the archive from growing forever, items will be removed after 10 years.

CHAPTER 6

Archive management

6.1 Automatic archive management

kopano-archiver can attach archive stores automatically, based on user attributes stored in LDAP or in Active Directory. When using this way of attaching stores, Kopano Archiver will create archive stores on the archive server and attach the user stores to these archive stores based on information found in LDAP or in Active Directory. When using this method of attaching and detaching Kopano WebApp will load archive stores automatically.

To use this feature, setting enable_auto_attach must be set to yes in /etc/kopano/archiver.cfg:

```
enable_auto_attach = yes
```

Alternatively kopano-archiver can be run periodically to perform the auto-attach operation:

```
kopano-archiver --auto-attach
```

6.1.1 Using OpenLDAP

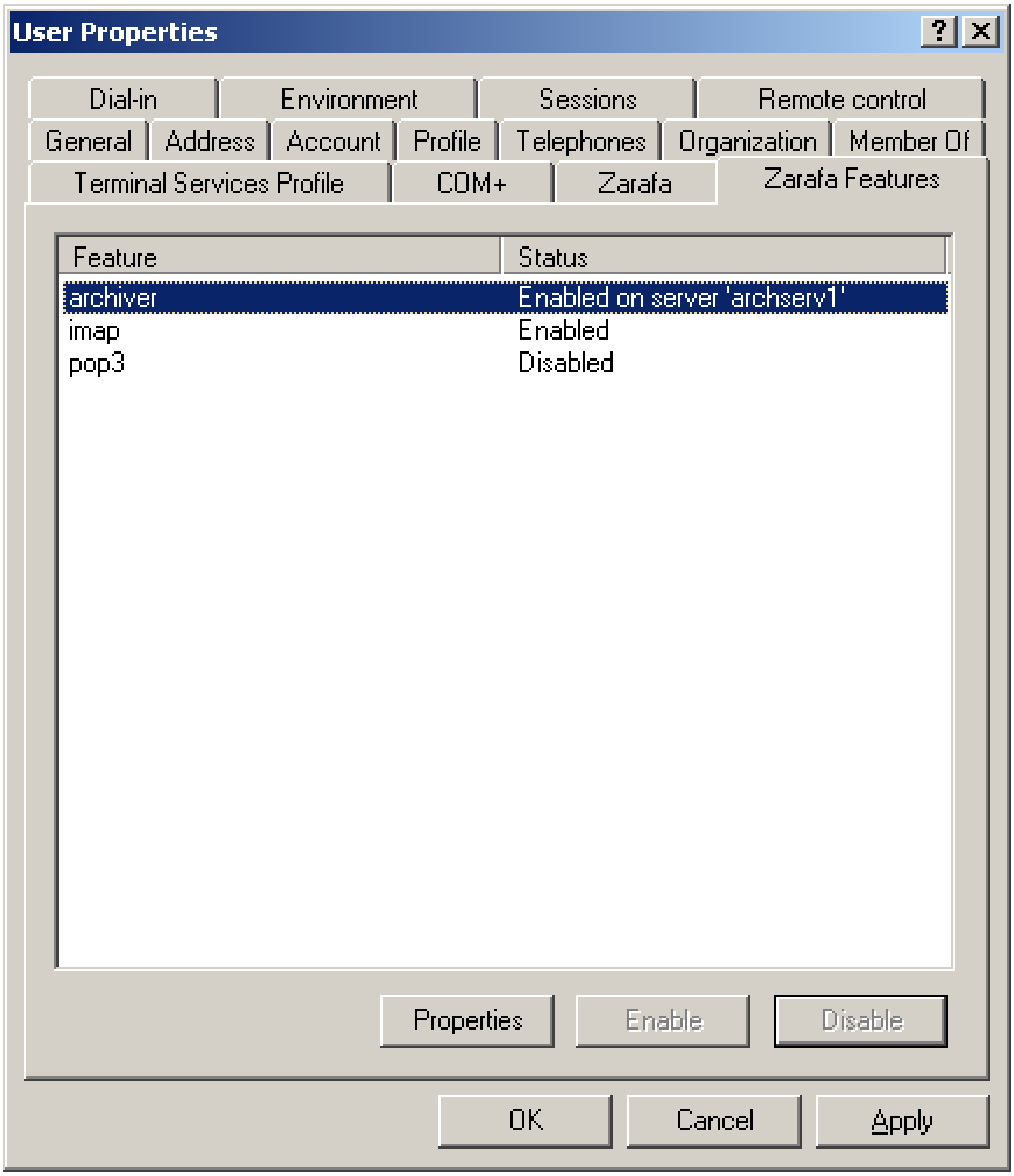
To add an archive store to a user through LDAP, attribute kopanoUserArchiveServers needs to be modified. This is a multi-value attribute, which needs to be set for the server name or server names of the servers that contain an archive store for the user.

A typical ldif (LDAP configuration file) would look like this:

Note: The archive store won't be created or attached until the next run of kopano-archiver -A with enable_auto_attach = yes or kopano-archiver --auto-attach.

6.1.2 Using Active Directory

To add an archive store to a user in Active Directory, one has to open the user in the *Active Directory Users and Computers* window and select the Kopano Features tab.



Feature	Status
archiver	Enabled on server 'archserv1'
imap	Enabled
pop3	Disabled

Figure 5.1. ADS Features tab

Next, select the `Archiver` feature and click `Properties`. This will pop up the dialog in which the server names of the servers on which an archive store should exist for the selected user or users.

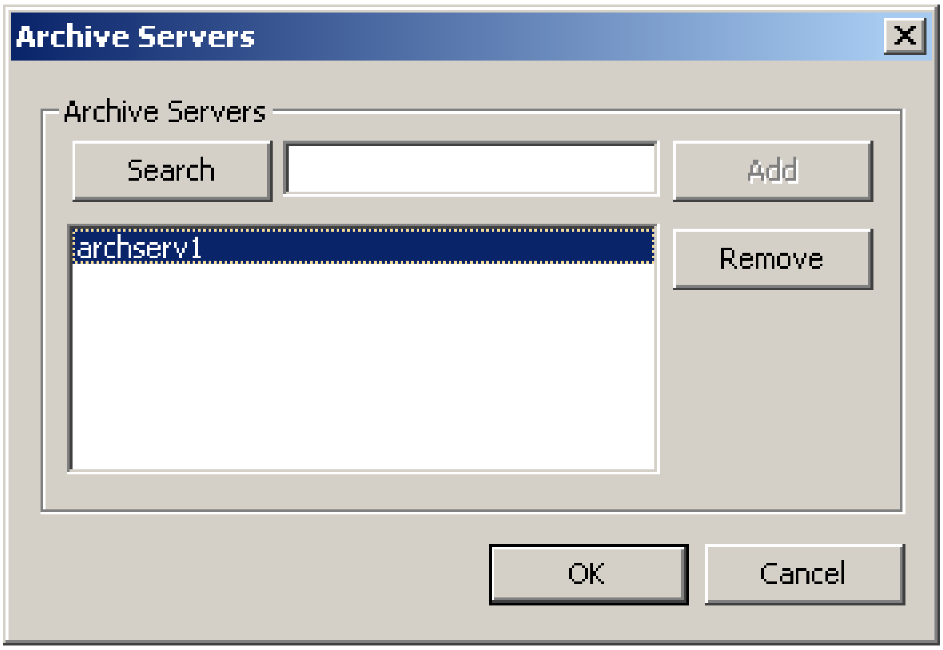


Figure 5.2. ADS Select archive servers

When a user has an automatic archive and the archiver feature in Active Directory is removed, the automatic archive will be automatically detached the first time an archive run is performed with auto-attach enabled. See also *Removing an archive store*.

Note: Archive stores won't be created or attached until the next run of `kopano-archiver -A` with `enable_auto_attach = yes` in its configuration file or by running `kopano-archiver --auto-attach`.

6.1.3 Listing attached archives

To see which archive stores are attached to a user's primary store execute the following command:

```
kopano-archiver -u <user name> --list
```

This will output a list of attached archives. Each line contains the name of the archive store, the name of the folder that acts as root for the archive and the access rights the owner has for the archive store.

The store name will equal the name that was passed when the archive was attached. The archive folder name will be one of three things:

1. it will be the full name of the user if no name was specified when the archive was attached
2. if a name was specified, the archive folder will be that name
3. if the archive was attached in a one-to-one configuration, it will be *Root Folder*.

The access rights will be *Read Only* when the archive was attached without write permissions. If it was attached with write access it will be *Read/write*. If the permissions were manually changed after the archive was attached, the rights field will display the configured role for the rights or all the rights if no matching role exists.

6.1.4 Detaching an archive

To detach an archive mailbox of a user, the archive server has to be removed in the Kopano feature tab.

Note: When detaching an archive that already contained archived and stubbed messages, the stubbed messages can still be opened.

6.1.5 Listing users that have an archive attached

To show which users have an archive attached execute the following command:

```
kopano-archiver --list-archiveusers
```

6.1.6 See details of an archive store

The details of an archive store can be obtained in two ways:

- By requesting the details of the user owning the archive:

```
# kopano-admin --details user1
Username:           user1
Fullname:           User 1
Emailaddress:       user1@cluster.sio2
Active:             yes
Administrator:      no
Address book:       Visible
Auto-accept meeting req:no
Home server:        cnode-1
Last logon:         12/07/2015 03:41:32 PM
Last logoff:        12/07/2015 03:41:32 PM
Mapped properties:
   PR_GIVEN_NAME           User
   PR_SURNAME              One
   PR_EC_ENABLED_FEATURES  pop3
   PR_EC_DISABLED_FEATURES imap
   PR_EC_ARCHIVE_SERVERS   cnode-2
Attached archives:  1
   Root Folder in Archive - User 1 [Read Only]
Quota overrides:    no
Warning level:      unlimited
Soft level:         unlimited
Hard level:         unlimited
Current store size: 14.86 MiB
Groups (1):
   Everyone

Archive details on node 'cnode-2':
Current store size: 114.68 MiB
```

All attached archive details are appended at the end.

- By explicitly requesting the archive details on a specific node:

```
# kopano-admin --details user1 --type archive --node cnode-2
Current store size: 114.68 MiB
```

Important: The node on which the archive is stored must be specified in order to find the archive.

These methods only apply to genuine archive stores. Details of regular stores that are manually attached as archives can be obtained by obtaining the details of the user owning that store:

```
# kopano-admin --details archive
Username:            archive
Fullname:            Archive Store
Emailaddress:        archive@cluster.sio2
Active:              no
Administrator:       no
Address book:        Hidden
Auto-accept meeting req:no
Home server:         cnode-2
Last logon:          12/09/2011 03:41:32 PM
Last logoff:         12/09/2011 03:41:32 PM
Mapped properties:
   PR_GIVEN_NAME            Archive
   PR_SURNAME               Archive
   PR_EC_ENABLED_FEATURES   pop3
   PR_EC_DISABLED_FEATURES  imap
Quota overrides:     no
Warning level:       unlimited
Soft level:          unlimited
Hard level:          unlimited
Current store size:  114.68 MiB
Groups (1):
   Everyone
```

6.1.7 Unhooking an archive store

An archive store can be unhooked the same way as a regular store, but with the addition of the `type` and `node` arguments:

```
# kopano-admin --unhook-store user1 --type archive --node cnode-2
Store unhooked.
```

6.1.8 Hooking an archive store

An archive store can be hooked the same way as a regular store, but with the addition of the `type` and `node` arguments:

```
# kopano-admin --list-orphans --node cnode-2
Stores without users:
   Store guid                          Guessed username    Last login      Store␣
↪size   Store type
   -------------------------------------------------------------------------------
↪---------------
   F1A6BFCD67604B0FB733F746F1D00A91    user1               <unknown>       0       ␣
↪     archive
# kopano-admin --hook-store F1A6BFCD67604B0FB733F746F1D00A91 -u user1 --type␣
↪archive --node cnode-2
Store hooked.
```

6.1.9 Removing an archive store

An archive store can be removed the same way as a regular store, but with the addition of the `type` and `node` arguments:

```
# kopano-admin --unhook-store user1 --type archive --node cnode-2
Store unhooked.
# kopano-admin --list-orphans --node cnode-2
```

```
Stores without users:
   Store guid                          Guessed username    Last login      Store␣
↪size  Store type
   --------------------------------------------------------------------------------
↪---------------
   F1A6BFCD67604B0FB733F746F1D00A91    user1               <unknown>       0      ␣
↪     archive
# kopano-admin --remove-store F1A6BFCD67604B0FB733F746F1D00A91 --node cnode-2
Store removed.
```

CHAPTER 7

Running the Archiver

7.1 Performing archiving task

Archiving can be done for all users, all users on one primary server or on a per-user basis.

7.1.1 From the command line

The following command performs an archive run for all users:

```
kopano-archiver -A
```

Passing option `--local-only` to `kopano-archiver` tells it to archive only the primary stores that live on the server to which kopano-archiver is connected. Unless configured otherwise in the configuration file, this is the server on which kopano-archiver is run:

```
kopano-archiver -A --local-only
```

To run `kopano-archiver` for a specific user use the -u option:

```
kopano-archiver -u <user name> -A
```

It is recommended to perform an Archiver run every night. This can be done by adding the following line to `/etc/crontab`:

```
0 1 * * * root [ -x /usr/sbin/kopano-archiver ] && /usr/sbin/kopano-archiver -A
```

7.2 Cleanup

To keep the archive synchronized with primary stores after items are deleted from the primary store, start the archiver in cleanup mode. This is a separate operation from archiving. Since cleanup may be a long process, it is recommended to run it less often than archive operations.

When `kopano-archiver` performs a cleanup run in which it encounters messages that exist in an archive and for which in main store a stub used to present which has now been deleted, it will remove the message from the archive. This is useful to avoid the situation that archived messages exist which have properties set that indicate that they are stubbed, while no corresponding stub exists in main(store.)

What happens to such archived messages depends on configuration:

1. cleanup_action is set to `delete`. In this case messages for which no stubs exist will be removed from the archive.
2. cleanup_action is set to `store`. In this case messages for which no stubs exist will be moved to a folder named `Kopano Archive\Deleted`.

Note: Archiver can remove *soft deleted* items from the main store (or stores) for which a cleanup run is performed. This avoids the situation in which *soft deleted* stubs may be restored for which the archived versions no longer exist. To enable this feature, configuration item `purge_soft_deleted` must be set to `yes`. See chapter *Softdelete restore* in the *Administrator Manual* for more information on *soft deleted* items.

7.2.1 From the command-line

The following command performs a cleanup for all users:

```
kopano-archiver -C
```

Passing option `--local-only` to `kopano-archiver` causes it to cleanup only archives of users who have a store on the server to which `kopano-archiver` is connected. This is the server on which `kopano-archiver` is executed unless otherwise configured in the configuration file.

```
kopano-archiver -C --local-only
```

It's also possible to explicitly specify which users archive to cleanup:

```
kopano-archiver -u <user name> -C
```

It's recommended to perform the cleanup run once a week. This can be done by adding the following line to `/etc/crontab`.

```
0 3 * * 0 root [ -x /usr/sbin/kopano-archiver ] && /usr/sbin/kopano-archiver -C
```

This will clean up all archives every sunday at 3:00 am.

CHAPTER 8

Configuration considerations

This chapter can be used as a guideline for configuration choices in the archiver rollout, based on the two scenarions of either storage or history based primary usage. All configuration options will be explained together with typical value for the setting.

- **Archive permissions**

When the archiver is deployed for storage optimalisation users will require the same permissions on the archive mailbox as on their primary mailbox. In a history based archive the permissions on the archive mailbox will be typically read-only, so the user can delete no items from the archive. With the Kopano Archiver the default permissions can be set when the a user is attached to an archive mailbox.

- **Copying time**

In a storage based archive the message will be typically archived after 30 or 60 days, so only the more important will be copied to the archive mailbox and non-important email will not take any space on the archive server as it will be removed prior to the archiving actions. In a history based archive the messages will be automatically archived at delivery time. In this case it's not necessary to set the archive time, as the messages are already in the archive store available. To use the history based archive feature both the delivery dagent and spooler need to have the option archive_enabled set to yes in order to store directly all emails at receive/sent time.

- **Stubbing time**

In both store and history based archives, stubbing will typically be set to 30-60 days. When choosing the right stubbing time, please keep in mind the body and attachments of stubbed emails can only be accessed when there is an online connection to the archive server. For this reason the stubbing time should not be set to low.

- **Deleting time**

As the deletion time will make items unavailable in the primary mailbox, the value will be typically have a value of 6 months or 1 year. After this time users normally don't open those emails that frequently anymore and in case the emails should be access, the archive mailbox can be added as a delegate mailbox. Having a high time for this setting will result in larger amount of items in a folder. When users should always access all items from the primary mailbox, the delete_enable setting setting of archiver should be set to No.

- **Purge time**

The purge setting will delete items completely from the archive mailbox, so emails will not be available at all. In case of a history based archive, so users will not be able to cleanup their own archive mailbox, this setting is recommended to use. When this setting is not configured, items will be never removed from the archive server. The typical time for this setting is 5-7 years, depending on your considerations and reasons for using the history based archive. When using the storage based archive the prefered time can be configured based on user experience and the average lifetime of items in mailboxes.

Note: This configuration option should be carefully considered including any legal aspects, as it will remove items from the email system.

- **Quota**

Like a normal installation of Kopano, archive servers can also have quotas configured. When using a history based archive setup a reached quota will stop the history archiving process, so in this case quota should not be set. In case of a storage based archive the archive is typically configured to reduce the data size on the primary server. When the quotas are set on the archive server the archive process will not be able to archive emails when the quota is reached and emails will be kept on the primary server, so therefore it's not recommended to configure quota on the archive.

- **Quota on primary server**

Also when the Kopano Archiver is integrated with a running KC environment mailbox quotas can be configured. When having the stubbing setting enabled, the stubbed items will only use around 1kB of size in the database.

CHAPTER 9

Archiver Extras

The Kopano Archiver Extras package contains additional tools to enhance basic Kopano Archiver functionality.

9.1 Installation

Kopano Archiver Extras can be found in the kopano-archiver-extra package.

9.2 The tools

9.2.1 Kopano Archiver ACL Sync

Description

`kopano-archiver-aclsync` synchronises archive ACL settings with those of the primary store.

When a user has set permissions for other users or groups on his or her store or folders, those other users will need at least read-permissions on this persons archive as well, so that they can read stubbed messages or access the archive directly. These permissions cannot be set by the owner of the archive when the archive was attached without write privileges. Even when another user has write permissions, it's a nuisance to set all the permissions twice, or possibly more often when multiple archives are attached.

Note that no user will ever get more rights on a store or folder than the owner of the archive. When the archive was attached without write permissions, no user will get write permissions on the archive stores. For every archived folder in an archived store `kopano-archiver-aclsync` will first determine the rights of the owner of the archive. After that it will get all the entries from the ACL of the current folder except those of the owner. Each right will be masked with the rights of the owner before being added to the ACL of the archive folder.

Usage

```
# kopano-archiver-aclsync [options] [users]

options:
-h serverpath            : Host to connect to.
-s sslkey_file           : SSL key file for authentication.
-p sslkey_pass           : Password for the SSL key file.
```

`users` is a space separated list of users for which to synchronize the ACL settings. If no user is specified all users will be processed.

9.2.2 Kopano Archiver ACL Set

Description

kopano-archiver-aclset sets or updates permissions on an archive store for the owner of that store.

When an archive is attached to a store, the owner of the store gets read-only or read/write permissions on the archive, depending on the configuration file or command line options used at the time of attaching. Also, when an archive is attached to a store of a non-active user, it is impossible to set permissions for that non-active user.

In the first case, it may be desirable to change the permissions at a later time because of company policy changes or because the original setting was wrong. In the second case, when the user is converted to an active user, permissions need to be reset because the owner won't have any permissions on his archive.

Usage

```
# kopano-archiver-aclset [OPTIONS] [users...]

options:
-h | --host         Host to connect to. Default: file:///var/run/kopano/server.sock
                    Three formats are allowed for this option:

                      UNIX socket : file://<path to the UNIX socket>
                             HTTP : http://<host or IP>:<port>/kopano
                      Secure HTTP : https://<host or IP>:<port>/kopano

-s | --sslkey-file  SSL key file for authentication.
-p | --sslkey-pass  Password for the SSL key file.
-w |                Grant write permissions on the archive.
--writable <y|n>    Enable or disable write permissions.
--help              Show this help message.
```

users is a space separated list of users for which to synchronize the ACL settings. If no user is specified all users will be processed.

Kopano-archiver-aclset should be executed on a regular basis from a cronjob to synchronise the rights between the primary store and the archive store .

9.2.3 Kopano Archiver Restore

Description

kopano-archiver-restore is a utility that restores archived messages from an archive. Use kopano-archiver-restore to destub all stubbed messages and restore all messages that were deleted after archiving. Do not drag and drop messages back from an archive to a main store in Kopano WebApp. Using kopano-archiver-restore ensures that restored messages are sanitised, so that they can be re-archived properly later.

Usage

```
Usage: kopano-archiver-restore [OPTIONS] user
OPTIONS:
-h | --host           : Host to connect to. Default: file:///var/run/kopano/server.
↪sock
-s | --sslkey_file    : SSL key file for authentication.
-p | --sslkey_pass    : Password for the SSL key file.
-l | --log-file       : Specify log file.
--detach              : Detach the selected or all archive stores before
```

```
                            starting the restore procedure. This avoids the
                            Archiver from rearchiving restored messages.
--unhook                    : Unhook the selected or all archive stores once
                            the restore process has completed. This implies
                            --detach and only works on archive stores.
--remove                    : Remove the selected or all archive stores once
                            the restore process has completed. This implies
                            --unhook and only works on archive stores.
-v | --verbose              : Increase console loglevel. Can be specified multiple
                            times.
-q | --quiet                : Decrease console loglevel. Can be specified multiple
                            times.
-N | --dry-run              : Don't actually modify anything.
--help                      : Show this help message.
```

Example

The following example will completely restore the store of john_doe and detaches and unhooks all archive stores while logging to `/tmp/john_doe_restore.log`

```
# kopano-archiver-restore --unhook -s /etc/kopano/ssl/archiver.pem -p password \
             -l /tmp/john_doe_restore.log john_doe
```

Note that no host is specified, causing kopano-archiver-restore to connect to `file:///var/run/kopano/server.sock`. The sslkey_file and sslkey_pass are specified in order to connect to the other nodes in the cluster.

CHAPTER 10

Archiver command line

The following table describes command line arguments and options that apply to Kopano Archiver. Please note that if no command line arguments are supplied the same help information is displayed as that from `--help`.

Table 9.1. Command line

Option	Alternative	Type	Meaning and usage
-u <name>		string	Select the archive of user <name>.
-l	–list	n/a	List archives for a specified user.
-L	–list-archiveusers	n/a	List users that have an archive attached.
-A	–archive	n/a	Perform archive operation. If no user is specified all user stores will be archived.
-C	–cleanup	n/a	Perform a cleanup of the archive stores attached to the user specified with `-u`. If no user is specified, all archives are cleaned up.
	–local-only	n/a	Archive or cleanup only those users that have their store on the server on which the archiver is invoked.
-a	–attach-to <archive store>	n/a	Attach an archive to the specified user. By default a subfolder will be created with the same name as the specified user. This folder will be the root of the archive.
-d	–detach-from <archive store>	n/a	Detach an archive from the specified user. If a user has multiple archives in the same archive store, the folder needs to be specified with –archive-folder.
-D	–detach <archive no>	n/a	Detach the archive specified by <archive no>. This number can be found by running `kopano-archiver -l`
	–auto-attach	n/a	When no user is specified with `-u`, all users will have their archives attached or detached based on the LDAP/ADS settings. When a user is specified only that user’s store will be processed. This option can be combined with `-A/--archive` to force an auto-attach run regardless of the enable_auto_attach configuration option.
-f	–archive-folder <name>	string	Specify an alternate name for the subfolder that acts as the root of the archive.
-s	–archive-server <path>	string	Specify the server on which the archive resides.
-N	–no-folder	n/a	Don’t use a subfolder that acts as the root of the archive. This implies that only one archive can be made in the specified archive store.
-w			Grant write permissions on the archive. This will override the `auto_attach_writable` config option.
	–writable	yes/no	Enable or disable write permissions. This will override the `auto_attach_writable` config option.
-c	–config	string	Use alternate config file.
	–help	n/a	Show a help message.

CHAPTER 11

Archiver configuration settings

The following table describes configuration settings that apply to Kopano Archiver. When settings marked `yes` in column *Required* aren't specified Archiver prints an error message and quits execution.

Table 10.1. Archiver settings

Setting	Type	Default	Required	Meaning and usage
server_socket	string		Yes	Location of server unix socket.
archive_enable	yes/no	yes	No	Setting this to `no` disables archiving.
archive_after	integer	30	No	Archive messages older than this number of days.
stub_enable	yes/no	no	No	Enables or disables stubbing.
stub_unread	yes/no	no	No	Enables or disables stubbing of unread messages.
stub_after	integer	0	No	Stub messages older than this number of days. When `archive_after` has a larger value than this setting, `archive_after` is used instead.
delete_enable	yes/no	no	No	Delete archived messages from the main store. Only archived messages can be deleted. Note that if this option is set to `yes`, no stubbing is performed.
delete_unread	yes/no	no	No	Enables or disables deleting unread archived messages from the main store.
delete_after	integer	0	No	Delete messages older than this number of days. When `archive_after` has a larger value than this setting, `archive_after` is used instead.
purge_enable	yes/no	no	No	Enables or disables purging of messages in the archive server(s).
purge_after	integer	2555 (7 years)	No	Purge messages older than this number of days. When `archive_after` has a larger value than this setting, `archive_after` is used instead.
cleanup_action	string	store	No	Specify what action should be taken on archive messages whose primary message has been deleted. Possible options are: `store`: the archived message in a special folder named `Kopano`

CHAPTER 12

Legal Notice

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.