



# **Open-Xchange™ Integration with eDirectory**

## **eDirectory as authentication source for Open-Xchange Server 5**

v0.90

Author: Stephan Martin

Contributors: Marcus Klein, Nicolas Barcet (AFOX), Farzad Farid (IDEALX),  
Dirk Wolter (Conet), Ralf Dobberschütz (Conet)

Editors: Robert Colombara

Layout: Robert Colombara

## Contents

<b>1.Overview.....</b>	<b>3</b>
1.1.Contribution.....	3
<b>2.Architecture Overview.....</b>	<b>4</b>
2.1.Open-Xchange Backends.....	4
2.1.1.Directory Service.....	4
2.1.2.E-Mail.....	5
2.1.3.Database.....	5
2.2.Open-Xchange Administration Framework.....	5
<b>3.eDirectory Integration - Concepts.....</b>	<b>7</b>
3.1.Simple Integration.....	7
3.2.Advanced Integration.....	8
<b>4.eDirectory Integration – Adaptions.....</b>	<b>10</b>
<b>5.Administering Users and Groups.....</b>	<b>12</b>

## 1. Overview

The Open-Xchange Server 5 is an Open Source based full featured E-Mail and Collaboration solution based on Linux.

The architecture of Open-Xchange Server 5 is completely based on open standards and open protocols to allow maximum flexibility regarding the integration in existing infrastructures.

This whitepaper describes how an Open-Xchange Server (OX Server) can be set up to use Novells eDirectory as authentication source for users and groups, as well as storage for addressbooks.

This document is meant to be used as a guide, a whitepaper that describes the concepts, and not as a complete How-To. Nevertheless some configuration files e.g. schema files will be made available in the Open Source Community wiki and linked within this document.

### **Attention:**

Deep knowledge about Linux services and administration as well as about directory services in general and especially eDirectory will be necessary to run an integration project like described below.

You should definitely know what you do and not just follow the documentaion. If you don't have the feeling, that you understand everything written below, it may be a good idea to contact an experienced Open-Xchange partner in your area.

### 1.1. Contribution

This whitepaper was written by Open-Xchange in cooperation with our Partners Conet in Germany and IDEALX in France as with support from the "Association Francophone pour le developpement d'Open-Xchange" (AFOX).

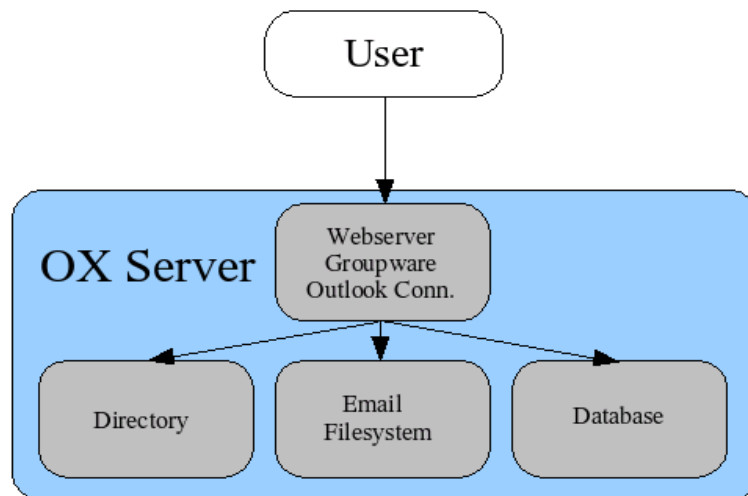
At this place we would like to thank everybody for the contribution and support.

## 2. Architecture Overview

This chapter gives a brief overview of the architecture of the Open-Xchange Server and its backends, as well as the different possibilities to achieve an integration into existing eDirectory infrastructures.

### 2.1. Open-Xchange Backends

The Open-Xchange server itself does not hold any information itself. All information is stored in different backend services. These services are selected to fit best to the kind of data which needs to be kept.



*Figure 1: Storage Subsystems inside of the OX Server*

#### 2.1.1. Directory Service

Storage of the OX user information needed for authentication is handled by a directory service.

The second usage of the directory is for storing the global and the private addressbooks. The global addressbook which is accessible by everybody and the private addressbooks, which are accessible only by the owner itself are exported to the LDAP server to allow access with standard clients like Mozilla Mail or Outlook Express.

The standard installation of the Open-Xchange server makes use of the free and stable OpenLDAP server, which is included in the underlying Linux distributions.

This is obviously the backend on which this paper will focus in detail.

### **2.1.2. E-Mail**

E-Mails are stored in an external E-Mail service. This service is accessed via the standard protocols IMAP and SMTP. The E-Mail components used in the standard setup are the Open Source MTA `postfix` and the Open Source IMAP server `cyrus-imapd`.

Both services are configured to obtain their information for authentication and mailrouting from the directory service, which will be mentioned in this paper as well.

During user creation normally the administration process creates the mailboxes of the users. This can be left for the webmail frontend, which can be configured to create the standard folders during the first login of the user.

### **2.1.3. Database**

All groupware data, appointments, tasks, etc. are stored in a database. The standard installation of Open-Xchange server makes use of the free and stable database PostgreSQL, which is included in the underlying Linux distributions.

The folder permissions for the groupware access are stored in the database. This is the reason, why user administration requires changes in the database as well.

## **2.2. Open-Xchange Administration Framework**

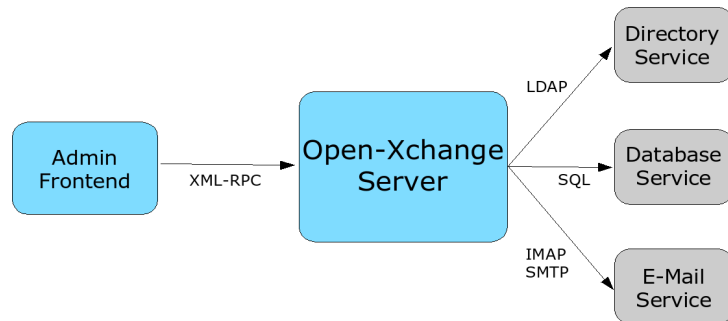
This chapter describes the standard setup to administer an Open-Xchange server with integration of all of its backends.

In the standard setup, one administration frontend is used to administer all users through the Open-Xchange server.

In the maintained version of the Open-Xchange server, this is done through the XML-RPC interface, which allows access to all administrative tasks. On the Open-Xchange server there is a daemon running, which will get all of these calls and distributes the necessary actions to the relevant backends.

This means in short: all data in the storage backend services is written by the Open-Xchange administration server itself.

The main advantage of this concept is, that the Open-Xchange server acts as an abstraction layer between the administration frontend and the backends. If one backend is exchanged, for example the database, only the relevant interface in the Open-Xchange server needs to be adapted and the other components don't need to be changed.



*Abbildung 2 Schema OX Administration Framework*

### **3. eDirectory Integration - Concepts**

This chapter describes two possible different levels of integration between the Open-Xchange server and Novells eDirectory.

What is common with both of the mentioned concepts is the prerequisite, that the Open-Xchange server will not be used as administration frontend anymore like it was described above.

If a company already uses an eDirectory for their identity management, it is very likely, that there are already frontends in place, which are not meant to be exchanged by the administration frontends from a groupware solution.

This means in detail, that these existing frontends of the directory service itself needs to be enhanced to:

- write the necessary information into the directory
- trigger the necessary actions for other backends in the Open-Xchange server

#### **3.1. Simple Integration**

The simple integration works without code changes to the Open-Xchange server and does not require the coding of special connectors inside eDirectory.

On the other hand it is necessary to enhance the eDirectory schema to implement this concept.

With this concept, there is one administration frontend, which writes the changes to user information directly into the directory server, it does not matter which frontend is used. It can be a custom frontend, which is already in place, it can be an extension to iManager, or it can be a simple LDAP browser with a template as well.

The Open-Xchange server will access this information from within the application and read the information. Write access from the Open-Xchange server to the directory is only necessary for password changes and for changes to the LDAP exported addressbooks.

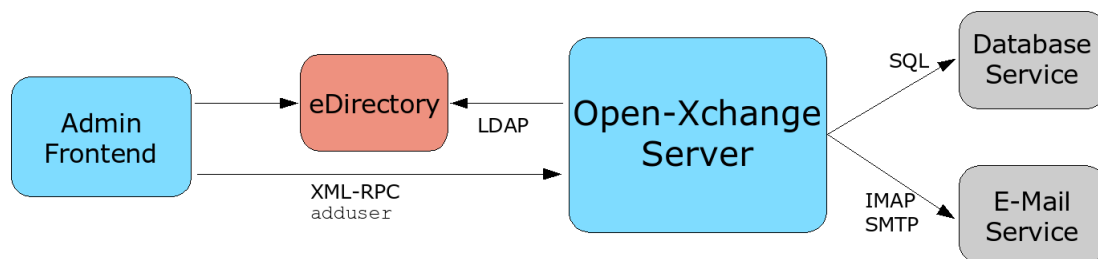


Abbildung 3 Schema Admin Framework eDirectory Integration - Simple

This setup means, that the changes to the directory and to the backends are separated, as not everything will be written through the Open-Xchange server anymore.

Therefore the existing administration frontend will take care to propagate the necessary changes to the Open-Xchange server, which itself takes care of propagating the changes to the other backends.

This can be achieved in several ways:

- The frontend itself can send an XML-RPC call to the Open-Xchange server
- The frontend can call the commandline tool `adduser`, which itself generates the appropriate XML-RPC call to the Open-Xchange server
- Another possibility is to run a regular job to analyse the content of the directory and to initiate the necessary actions, if new users are found.

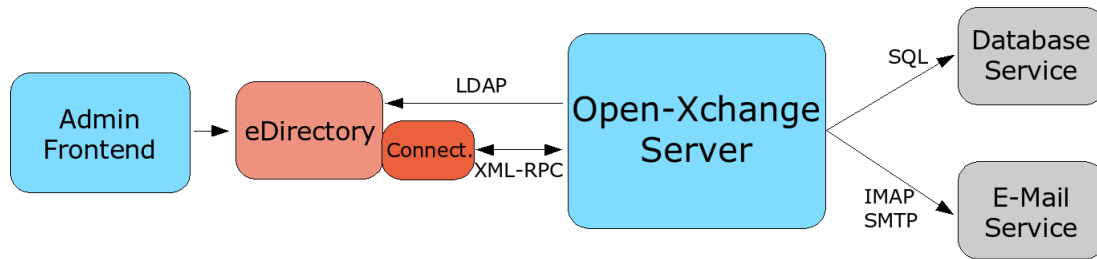
There is plenty room and flexibility to design the connection in a way, which fits best to the existing environment.

### 3.2. Advanced Integration

A more advanced way to integrate an Open-Xchange server into an eDirectory environment needs some more effort.

Basically it is the same like the one described above. The main difference is, that there is no split for the two channels of administration anymore. The only point of administration is writing into the eDirectory.

In eDirectory a connector can be implemented, which captures the changes in the user objects and triggers the necessary actions through the Open-Xchange server with XML-RPC calls.



*Illustration 4 Schema Admin Framework eDirectory Integration - Advanced*

## 4. eDirectory Integration – Adaptions

This chapter describes the necessary steps, which need to be done to achieve the integration described above.

As every implementation of an eDirectory may look different, this is no complete HowTo, but a high level description of the necessary tasks.

Example configuration files and schema files can be found in the Open Source community wiki.

1. Enhance the eDirectory schema:
  - Upload the schema files from the Open-Xchange wiki in the correct order through ICE or the corresponding webfrontend in iManager.
  - The given schema files will only be sufficient, if you are running eDirectory on Linux (OES) or if you have added the UNIX schema files for Linux User management (LUM) manually.
  - This task of enhancing the eDirectory schema should only be done by experienced eDirectory administrators.
2. Configure the E-Mail backends and the Open-Xchange server to access the new eDirectory LDAP server. This is done in the file:
 

```
/etc/openldap/ldap.conf
```
3. The LDAP interface of the groupware needs to be adapted to place the private addressbooks at a special subtree and not below the corresponding user object due to a restriction of the underlying X.500 schema in eDirectory, which does not allow the useage of an user object as container.
  - Additionally it is recommended to disable the export of the address entries to LDAP in the Open-Xchange LDAP interface, if no external LDAP address client is in use.
4. The permissions of the eDirectory need to be adapted:
  1. For “anonymous” binds there needs to be a system user, which is able to search all users and groups in the tree.
    - This user is configured in the file `/etc/openldap/ldap.conf`

2. Every user needs to be able to read the relevant entries from all other users for different reasons:
  1. Select the other users as participants for appointments, tasks, etc...
  2. Read the other users objects as addressbook entry, e.g. for sending emails
3. Every user needs read access to all group objects to select users from these groups
4. If the private addressbook export to LDAP shall be used, every user needs read and write access to every object below his private addressbook container.
5. The mapping of several attributes, the search bases and the search scopes for several queries need to be adapted in the Open-Xchange LDAP interface.

For example the usernames are represented in the attribute "cn" in eDirectory in opposite to the attribute "uid" in the standard OpenLDAP deployment.

This configurations options are set in the file:

`/opt/openexchange/etc/groupware/ldap.properties`

For this file an example is available in the Open-Xchange wiki as well.

6. The location of the private addressbook tree needs to be placed in a separate subtree. This is because the X.500 schema of eDirectory does not allow to use the users object as a container.

Another possibility is to switch of the export of the private addressbooks to LDAP. This saves a lot of useless entries in the directory server, if this feature is not used by the clients.

Both options can be set in the file:

`/opt/openexchange/etc/groupware/ldap.properties`

## 5. Administering Users and Groups

The user administration through eDirectory is straight forward. In the simple integration, mentioned above, there are three steps involved:

### 1. User Creation

The user will be created within eDirectory in the same way like every other eDirectory user without Open-Xchange integration. This can be done with Console One, iManager or every other frontend or integrated software, which is in use to administer the users in the existing eDirectory.

### 2. User Object Enhancement

The users object needs to be enhanced with the Open-Xchange objectclass and some mandatory attributes.

If any attributes are requested from Open-Xchange, which are available already in the directory through any other application, it is possible to change the attribute mapping for Open-Xchange in the file `ldap.properties`, so that it is not necessary to keep the same information in different attributes.

It is recommended to add at least the following attributes.

#### 1. Objectclass:

1. `OXUserObject`

#### 2. Attributes:

1. `OXTimeZone` (e.g. Europe/Berlin)

2. `smtpServer` (e.g. 127.0.0.1)

3. `imapServer` (e.g. 127.0.0.1)

4. `mailenabled` (e.g. OK)

5. `preferredLanguage` (e.g. DE)

6. `userCountry` (e.g. DE)

7. `maildomain` (e.g. test.tux)

8. `OXTaskDays` (e.g. 5)

9. `OXAppointmentDays` (e.g. 5)

### 3. Groups

Groups are handled just like normal groups within eDirectory. There is nothing special to take care of. Adding and removing users from groups works like expected.

(One pitfall: there are many programs under Linux, which will fall into trouble, if a user is member of more than 64 groups.)

## 4. Backend Deployments

### 1. Database

When the user is created in the directory service, some associations have to be created in the database as well to create standard folders, grant permissions, etc...

This can be done with the commandline tool:

```
adduser --sql ...
```

Another possibility is to communicate directly with the XML-RPC administration interface and trigger the necessary actions:

XXXXX

### 2. IMAP Mailbox

The user needs a mailbox with several folders, where the emails are physically stored. This mailbox needs to be created before it can be used. There are two different possibilities to achieve this:

#### 1. Creating the mailbox during user creation:

For this way, the same possibilities exist, like described for the database above:

This can be done with the commandline tool:

```
adduser --imap ...
```

Another possibility is to communicate directly with the XML-RPC administration interface and trigger the necessary actions:

XXXXX

#### 2. Let the mailbox and folders create during first usage/login through the user:

In `imapd.conf` the `autocreatequota` parameter has to be not 0 to allow cyrus to create the mailbox during the first login of the user.

In `webmail.properties` the parameter:

```
user.default.folder.autocreate
```

has to be set to `true`, to create the system folders through webmail during the first login of the user.