univention
be open.

OX Connector app

Release 3.2.2

Univention GmbH

Jun 25, 2026

CONTENTS

Welcome to the documentation about the **OX Connector** app. The app installs a connector to provision selected users and groups to a remote OX App Suite installation through the OX SOAP API. The app **doesn't** install OX App Suite to UCS (Univention Corporate Server).

This document addresses system administrators, who:

- operate UCS and OX App Suite.
- want to centrally manage users and groups in UCS.
- want to provision permitted users to OX App Suite.

This document covers the following topics:

1. *Installation* (page 3) about prerequisites and how to install with web browser and command-line.
2. *Configuration* (page 15) with a reference list about the app settings of the OX Connector app.
3. *Architecture* (page 23) of the app, how the connector works and the connector cache.
4. *Limitations* (page 27) of the app.
5. *Troubleshooting* (page 29) about log files, health check, queuing and rebuild the cache.
6. *Changelog* (page 39) about what changed in the different app versions.

This document doesn't cover the following topics:

- Installation, setup and usage of OX App Suite, see *App Suite Admin Guide 7.10.5* [1].
- Installation, setup and usage of UCS, see *UCS 5.2 Manual* [2].

To understand this document, you need to know the following concepts and tasks:

- Use and navigate in a remote shell on Debian GNU/Linux derivative Linux distributions like UCS. For more information, see Shell and Basic Commands[1] from *The Debian Administrator's Handbook*, Hertzog and Mas [3].
- Manage an app through Univention App Center[2] in *UCS 5.2 Manual* [2].

Your feedback is welcome and highly appreciated. If you have comments, suggestions, or criticism, please send your feedback[3] for document improvement.

[1] https://www.debian.org/doc/manuals/debian-handbook/short-remedial-course.en.html#sect.shell-and-basic-commands
[2] https://docs.software-univention.de/manual/5.2/en/software/further-software.html#computers-softwareselection
[3] https://www.univention.com/feedback/?ox-connector=generic

CHAPTER

ONE

INSTALLATION

The **OX Connector** app connects the UCS identity management with the OX App Suite database. For more information about how it works, see *How the connector works* (page 24).

1.1 Prerequisites

Before you as administrator can install the **OX Connector** app, you need to make sure that your environment fulfills the prerequisites.

1.1.1 OX App Suite server

For the *OX App Suite* server, you must ensure the following prerequisites:

1. The environment requires an installed OX App Suite instance. This documentation assumes that an OX App Suite installation already exists.

   For limitations about the **OX App Suite** app from Univention App Center and the connector, see *Integration of OX Connector and OX App Suite app* (page 27).

   For installation of OX App Suite, see *App Suite Admin Guide 7.10.5* [1].

2. The OX App Suite instance must allow SOAP requests, so that the UCS system, where the *administrator installs the OX Connector app* (page 4), can connect to `/webservices`.

3. You have to set up an administrator user in OX App Suite that can create OX contexts.

   The OX Connector can manage OX contexts. The installation of the **OX Connector** app needs username and password for that user and references them in the setting *OX_MASTER_ADMIN* (page 16) and *OX_MASTER_PASSWORD* (page 16).

   For manually managing OX contexts without the OX Connector, see *Contexts* (page 7).

4. Since version 2.2.0, OX must allow the use of duplicated *displaynames*. To enable it, add the following lines to the `user.properties` file.

   ```
   com.openexchange.user.enforceUniqueDisplayName=false
   com.openexchange.folderstorage.database.preferDisplayName=false
   ```

   OX must also allow all group names that you can enter in UDM. For that, add the following line to the `Group.properties`.

   ```
   CHECK_GROUP_UID_FOR_NOT_ALLOWED_CHARS=false
   ```

   Note

   This is configured by default in version 7.10.6-ucs7 of **OX App Suite** from the App Center.

1.1.2 UCS domain

Another prerequisite needs some steps in the UCS domain. To use the **OX Connector** app, the central LDAP directory needs the *referential integrity* overlay enabled. The overlay ensures that UDM objects provided by the OX Connector keep their integrity and always reference user objects correctly in the LDAP directory.

OX Connector on Primary Directory Node

If you install **OX Connector** on Primary Directory Node, the app already takes care of the necessary step. No further action required.

OX Connector on other system roles

If you install **OX Connector** on other UCS system roles[4] than the Primary Directory Node, you need to run the following commands:

Listing 1.1: Activate OpenLDAP *referential integrity* overlay on Primary Directory Node.

```
$ ucr set ldap/refint=true
$ service slapd restart
```

For more information about the *referential integrity* overlay, see *Referential Integrity in OpenLDAP Software 2.4 Administrator's Guide* [4].

1.2 Installation on UCS system

As administrator, you can install the **OX Connector** app like any other app with Univention App Center. Make sure to fulfill the *Prerequisites* (page 3).

UCS offers two different ways for app installation:

- With the web browser in the UCS management system
- With the command-line

For general information about Univention App Center and how to use it for software installation, see Univention App Center[5] in *UCS 5.2 Manual* [2].

1.2.1 With the web browser

To install **OX Connector** from the UCS management system, use the following steps:

1. Use a web browser and sign in to the UCS management system.
2. Open the *App Center*.
3. Select or search for *OX Connector* and open the app with a click.
4. To install the OX Connector, click *Install*.
5. Adjust the *App settings* to your preferences. For a reference, see *Configuration* (page 15).
6. To start the installation, click *Start Installation*.

Note

To install apps, the user account you choose for login to the UCS management system must have domain administration rights, for example the username `Administrator`. User accounts with domain administration rights belong to the user group `Domain Admins`.

[4] https://docs.software-univention.de/manual/5.2/en/domain-ldap/system-roles.html#system-roles
[5] https://docs.software-univention.de/manual/5.2/en/software/app-center.html#software-appcenter

For more information, see Delegated administration for UMC modules[6] in *UCS 5.2 Manual* [2].

1.2.2 With the command-line

To install the **OX Connector** app from the command-line, use the following steps:

1. Sign in to a terminal or remote shell with a username with administration rights, for example `root`.
2. Adjust the settings to your preferences with the appropriate installation command. For a reference, see *Configuration* (page 15). To pass customized settings to the app during installation, see the following command template:

```
$ univention-app install ox-connector --set $SETTING_KEY=$SETTING_VALUE
```

Example:

```
$ univention-app install ox-connector --set \
  OX_MASTER_ADMIN="oxadminmaster" \
  OX_MASTER_PASSWORD="some secure password" \
  LOCAL_TIMEZONE="Europe/Berlin"` \
  OX_LANGUAGE="de_DE" \
  DEFAULT_CONTEXT="10" \
  OX_SMTP_SERVER="smtp://my-smtp.example.com:587" \
  OX_IMAP_SERVER="imap://my-imap.example.com:143" \
  OX_SOAP_SERVER="https://my-ox.example.com"
```

Note

The installation process asks for the password of the domain administrator `Administrator`. To use another username and password for installation, pass different values with the options `--username` and `--pwdfile`. For more information, see **`univention-app install -h`**.

[6] https://docs.software-univention.de/manual/5.2/en/central-management-umc/delegated-administration.html#delegated-administration

CHAPTER

TWO

USAGE

The **OX Connector** centrally manages users, groups, OX contexts, OX access profiles and functional accounts with the web based management system in UCS. This section shows how.

To follow the tasks, you need to sign-in to Univention Management Console (UMC) with a user account with domain administration rights. For more information, see Delegated administration for UMC modules[7] in *UCS 5.2 Manual* [2].

2.1 Contexts

OX App Suite uses *contexts* to collect users, groups, and resources for collaboration in a virtual space. Data from one context isn't visible to other contexts. For more information about contexts, see *App Suite Context management* [5].

To view, add, update, or delete a context, you navigate to *Domain ‣ OX Contexts* in UMC.

Note

If you don't want the OX Connector to manage *contexts*, you can manually manage them in OX App Suite, as long as you maintain the *context* configuration for the OX Connector in the `/var/lib/univention-appcenter/apps/ox-connector/data/secrets/contexts.json`.

This approach doesn't require to share the credentials for the OX context administrator.

2.2 Users

To enable users for OX App Suite, administrators can either create user accounts or update existing ones.

To enable a user account for OX App Suite, run the following steps:

1. Navigate to *Users ‣ Users* in UMC and click to open.

Add user account

To create a user account:

2. Click *Add* to create a user account and select the *User template* `open-xchange groupware account`.
3. Click *Next*.
4. Fill out the required fields. To fill out more attributes, click *Advanced*.
5. When finished, click *Create user*.

[7] https://docs.software-univention.de/manual/5.2/en/central-management-umc/delegated-administration.html#delegated-administration

Update user account

To update a user account:

2. Click the username for the user you want to update.
3. Go to the *Apps* tab and activate the *Open-Xchange* checkbox. The tab *Open-Xchange* appears.
4. Define an email address for the user at *General ‣ Primary e-mail address (mailbox)*.
5. Click *Save*.

See also

User management[8] in *UCS 5.2 Manual* [2].

2.3 Groups

The **OX Connector** app adds a group to the same context as the group members. When the last group member leaves the group, the connector removes the group from OX App Suite.

To enable a group for OX App suite, run the following steps:

1. Navigate to *Users ‣ Groups* in UMC and click to open.

..tab-set:

```
.. tab-item:: Add group

   To create a group:

   2. Click :guilabel:`Add` to create a group.

   #. On the *General* tab, fill out the required fields and add users as group
      members.

   #. Go to the *OX App Suite* tab and activate the *Activate Group in OX*.

   #. Click :guilabel:`Create group`.


.. tab-item:: Update group

   To update a group:

   2. Click a group to edit.

   #. The UDM module *Groups* automatically enables *Activate Group in OX*, when
      you edit a group. UMC displays a notification.

      If you don't want to enable the group, clear the checkbox *Activate Group
      in OX* on the *OX App Suite* tab.

   #. Click :guilabel:`Save`.

   .. warning::

      When you as administrator update a group, that already is a group in OX App
```

(continues on next page)

[8] https://docs.software-univention.de/manual/5.2/en/user-management/index.html#users-general

(continued from previous page)

```
      Suite, and you clear the checkbox *Activate Group in OX* on the *OX App
      Suite* tab, the connector removes this group from OX App Suite.

   To update a group from the command-line, run the following command:

   .. code-block:: console

      $ udm groups/group modify --dn $dn_of_group --set isOxGroup=OK

.. tab-item:: Remove group

   To remove a group from OX App Suite:

   2. Click a group to edit.

   #. Go to the *OX App Suite* tab and clear the checkbox *Activate Group
      in OX*.

   #. Click :guilabel:`Save`.

   To remove the group from OX App Suite through command-line, run the following
   command:

   .. code-block:: console

      $ udm groups/group modify --dn $dn_of_group --set isOxGroup=Not
```

See also

Group management[9] in *UCS 5.2 Manual* [2].

2.4 Access profiles

The OX Connector already provides ready-to-use *access profiles* for OX App Suite users. Administrators can create custom *access profiles* in UMC in the *LDAP directory* module at *Domain ▸ LDAP directory* at the directory location `open-xchange/accessprofiles/`.

For limitations about plausibility verification, see *No plausibility validation in access profile rights* (page 28).

2.5 Functional accounts

Added in version 2.0.0.

OX App Suite shares functional mailboxes among other users in the same context.

With the UDM (Univention Directory Manager) module `oxmail/functional_account` administrators can add, update or delete objects for functional accounts. OX App Suite users with the same functional account share the read status. Emails to addresses of functional accounts show up in the OX Mail view for every user where administrators granted the permission.

Warning

Open-Xchange marked this feature as deprecated in favor of *Shared accounts* (page 10).

[9] https://docs.software-univention.de/manual/5.2/en/groups.html#groups

2.5.1 Default LDAP position for functional accounts

Added in version 2.2.12.

When you create a new `oxmail/functional_account` object in UMC (Univention Management Console) the default position for these new objects in the directory tree is `cn=functional_accounts,cn=open-xchange,$LDAP_BASE`.

However, you can add additional default containers for the `oxmail/functional_account` so that UMC will ask for a position before creating the new object.

In the UMC module *LDAP directory* open the container `univention` in the tree view (left) and then open the object `default containers` in the object list (right). Click on `OX App suite` and add additional default containers to the list of `Default container for OX functional accounts`. The values are LDAP DNs of existing container objects in your LDAP directory, which must include the LDAP base DN.

2.6 Resources

OX App Suite uses *OX Resources* to manage resources like rooms or equipment that users can book for appointments. For more information about resource management, see *App Suite Resource management* [6].

To view, add, update, or delete a resource, you navigate to *Domain* ‣ *OX Resources* in UMC.

2.7 Shared accounts

Added in version 3.2.0.

OX App Suite lets users and groups access shared accounts. Users with a shared account can read its email and calendar entries. As an administrator, you can configure fine-grained permissions for users and groups. The OX Connector app provides UDM modules to manage shared accounts and the permissions of users and groups.

Important

The *Shared accounts* feature requires *OX App Suite* version 8.49 or later. A runtime check deactivates the feature when *OX App Suite* doesn't support shared accounts.

See also

Shared accounts[10]

2.7.1 UDM module for shared accounts

As an administrator, you can use the UDM module `oxmail/shared_account` to add, update, or delete objects for shared accounts and manage their permissions. You can find the UDM module in the *Management UI* under *LDAP directory* at the directory location `open-xchange/shared_account`.

Every `oxmail/shared_account` object contains a list of users and groups with their respective permissions. Each user and group entry in the list links to an `oxmail/shared_account_permissions` object.

See also

LDAP directory module[11]
for information about the *LDAP directory* management module.

[10] https://documentation.open-xchange.com/8/middleware/permissions_and_capabilities/shared_accounts.html
[11] https://docs.software-univention.de/nubus-manual/1.x/en/management/domain/ldap.html#nubus-domain-ldap

2.7.2 UDM module for permissions

OX App Suite uses permission objects to control user and group access to shared accounts. OX Connector provides ready-to-use *permissions* for OX App Suite shared accounts, including *Full Calendar Access*, *Full Mail Access*, *Full Mail and Calendar Access*, and *Read-Only Mail Access*. You can also create permissions to meet your requirements.

As an administrator, you can use the UDM module `oxmail/shared_account_permissions` to create, update, or delete permissions for shared accounts. You can find the UDM module in the *Management UI* under *LDAP directory* at the directory location `open-xchange/shared_account_permissions`.

When you create an `oxmail/shared_account` object, you can grant permissions to users and groups in UMC.

2.7.3 Migration from functional accounts to shared accounts

Added in version 3.2.1.

The shared accounts feature in OX App Suite deprecates the old functional accounts. OX Connector provides a script that lets you migrate from functional accounts to shared accounts.

Before you run the script, **`Dovecot`** must use the email address as the unique identifier for the mail accounts.

Danger

If your **`Dovecot`** installation uses a unique identifier other than the email address, **don't run** the migration script. In that case, the script deletes your functional accounts and creates shared accounts without their content.

Test the migration script and carefully review the results before you use it in production. The `dry-run` option runs the migration script without actually writing changes to OX App Suite, and prints statements from the steps during the migration.

For information about the migration script parameters, use the `--help` option. It provides options about addressing multiple functional accounts with one run, or providing credentials through environment variables.

For troubleshooting, see *Troubleshooting migration of functional accounts to shared accounts* (page 36).

Depending on your deployment of the OX Connector, choose one of the following options to run the migration.

App in Univention App Center

Run the migration script on Nubus for UCS on the system that has the OX Connector installed. Use the commands in Listing 2.1 and Listing 2.2. In the listing you need to provide the values for the following inputs:

`UDM_USERNAME`
: The username for the UDM user. The user account must be a member of the Authorization groups[12] in the *UDM HTTP REST API*.

`UDM_PASSWORD`
: The password for the `UDM_USERNAME`.

`REST_API_HOSTNAME`
: The FQDN of the UDM HTTP REST API in your domain.

`OPTIONAL_DESTINATION`
: The container for the shared account that the migration script creates.

`DESTINATION_OX_CONTEXT`
: The OX Context where the shared account will be created.

[12] https://docs.software-univention.de/nubus-customization/latest/en/api/udm-rest.html#customization-api-udm-rest-auth-group

Listing 2.1: Prepare migration to shared accounts

```
$ export UDM_USERNAME="<your-udm-user>"
$ export UDM_PASSWORD="<your-udm-password>"
$ export REST_API_HOSTNAME="<your-udm-rest>"
$ export OPTIONAL_DESTINATION="<optional-custom-destination-for-single-migration>"
$ export DESTINATION_OX_CONTEXT=<your-ox-context-id>
```

Listing 2.2: Run the migration from functional accounts to shared accounts

```
$ univention-app shell \
   ox-connector \
   /usr/local/share/ox-connector/resources/migrate_fupo_to_shared_account.py \
   "cn=example_fupo,cn=functional_accounts,cn=open-xchange,$(ucr get ldap/base)" \
   "Full Mail Access" \
   "$OPTIONAL_DESTINATION" \
   "$UDM_USERNAME" \
   "$UDM_PASSWORD" \
   "https://$REST_API_HOSTNAME/univention/udm" \
   --ox-context $DESTINATION_OX_CONTEXT
```

Consumer in Nubus for Kubernetes

To run the migration script in your Nubus for Kubernetes environment, use the following steps.

1. To configure the namespaces for your Nubus for Kubernetes environment and the OX Consumer deployment, set the environment variables as shown in Listing 2.3.

   NAMESPACE_N4K
   : The Kubernetes namespace for your Nubus for Kubernetes deployment.

   RELEASE_N4K
   : The release name for your Nubus for Kubernetes deployment. To list the release names in your namespace, run the command in Listing 2.4.

   NAMESPACE_CONNECTOR
   : The Kubernetes namespace of your OX Connector deployment. Typically, it's the same namespace as for Nubus for Kubernetes.

   Listing 2.3: Set environment variables for Nubus for Kubernetes and the OX Connector.

   ```
   $ export NAMESPACE_CONNECTOR="<your-namespace-for-the-connector>"
   $ export NAMESPACE_N4K="<your-namespace-for-nubus-for-kubernetes>"
   $ export RELEASE_N4K="<Release-name-for-nubus-for-kubernetes>"
   ```

   Listing 2.4: Show the release names in the namespace of Nubus for Kubernetes

   ```
   $ helm --namespace "$NAMESPACE_N4K" list -q
   ```

2. Retrieve the LDAP base DN from your Nubus for Kubernetes environment.

   You need the LDAP base DN of your Nubus for Kubernetes deployment. You provided the LDAP base DN during the deployment of Nubus for Kubernetes[13] in your `custom_values.yaml`. To retrieve the LDAP base DN, run the command in Listing 2.5.

   LDAP_BASE
   : The LDAP base DN of your directory service.

[13] https://docs.software-univention.de/nubus-kubernetes-operation/latest/en/deploy.html#nubus-deployment-all-deps

Listing 2.5: Retrieve parameters from Nubus for Kubernetes environment

```
$ export LDAP_BASE="$(kubectl \
   --namespace "$NAMESPACE_N4K" \
   get configmap \
   "$RELEASE_N4K-ldap-server" \
   -o "jsonpath={.data.LDAP_BASE_DN}")"
```

3. Configure the remaining parameters for the migration script.

   UDM_USERNAME
   : The username for the UDM user. The user account must be a member of the Authorization groups[14] in the *UDM HTTP REST API*.

   UDM_PASSWORD
   : The password for the `UDM_USERNAME`.

   FUNCTIONAL_ACCOUNT
   : The LDAP distinguished name (DN) of the functional account that you want to migrate.

   DESTINATION_OX_CONTEXT
   : The OX Context where the shared account will be created.

Listing 2.6: Define the remaining parameters for the migration

```
$ export UDM_USERNAME="<your-udm-user>"
$ export UDM_PASSWORD="<your-udm-password>"
$ export FUNCTIONAL_ACCOUNT="cn=example_fupo,cn=functional_accounts,cn=open-
↪xchange,$LDAP_BASE"
$ export DESTINATION_OX_CONTEXT=<your-ox-context-id>
```

4. Run the migration script. The example command in Listing 2.7 uses the variables that you defined in the previous steps.

   Verify the migration result before you continue with production use.

Listing 2.7: Run the migration script

```
$ kubectl \
   --namespace="$NAMESPACE_CONNECTOR" \
   exec ox-connector-0 -c main -- /bin/bash \
   -c 'UDM_USERNAME='$UDM_USERNAME' \
   UDM_URL="http://nubus-udm-rest-api:9979/univention/udm/" \
   UDM_PASSWORD='$UDM_PASSWORD' python3 \
   /usr/local/share/ox-connector/resources/migrate_fupo_to_shared_account.py \
   '$FUNCTIONAL_ACCOUNT' \
   "Full Mail Access" \
   --ox-context '$DESTINATION_OX_CONTEXT''
```

[14] https://docs.software-univention.de/nubus-customization/latest/en/api/udm-rest.html#customization-api-udm-rest-auth-group

CHAPTER

THREE

CONFIGURATION

The following reference shows the available settings for the **OX Connector** app.

3.1 App Settings

OX_SOAP_SERVER

Defines the server that has *OX App Suite* installed. Provide the protocol and the FQDN, for example `https://ox-app-suite.example.com`.

OX_SOAP_SERVER (page 15) instructs the OX Connector app in the Docker container, where it must look for the *OX App Suite* system. The Docker container must resolve the FQDN.

Required	Type	Initial value
Yes	String	`https://$hostname.$domainname`

For secure connections with HTTPS the Docker container needs to validate the certificate.

Note

If the *OX App Suite* instance uses a self-signed certificate or a certificate it can't validate, the OX Connector container needs the root certificate for validation. You need to store the self-signed certificate files in the `/var/lib/univention-appcenter/apps/ox-connector/data/conf/ca-certificates/` directory. For details, see *Import additional CA certificates* (page 22).

OX_IMAP_SERVER

Defines the default IMAP server for new users, if not explicitly set at the user object.

Required	Type	Initial value
Yes	String	`imap://$hostname.$domainname:143`

OX_SMTP_SERVER

Defines the SMTP server for new users, if not explicitly set at the user object.

Required	Type	Initial value
Yes	String	`smtp://$hostname.$domainname:587`

DEFAULT_CONTEXT

Defines the default context for users. The OX Connector doesn't create the `DEFAULT_CONTEXT` automatically. You as administrator must ensure, the default context exists before the OX Connector provisions the first user. To create a context, see *Contexts* (page 7).

Required	Type	Initial value
Yes	Integer	`10`

OX_LANGUAGE

Defines the default language for new users

Required	Type	Initial value
Yes	String	`de_DE`

LOCAL_TIMEZONE

Defines the default timezone for new users

Required	Type	Initial value
Yes	String	`Europe/Berlin`

OX_MASTER_ADMIN

Defines the user for the *OX App Suite* administrator user, also called *OX Admin user*. This user can create, modify, and delete contexts. The user must already exist. The administrator defines the username for the *OX Admin user* during the installation of OX App Suite.

Required	Type	Initial value
Yes	String	`oxadminmaster`

OX_MASTER_PASSWORD

Defines the password for the *OX Admin user*.

Required	Type	Initial value
No	Password	N/A

OX_IMAP_LOGIN

Defines the value that is used by OX to log in to the user's inbox. If this value is empty it is set to the user's mail address.

Required	Type	Initial value
No	String	N/A

In cases where you use single sign-on, you need to append this variable with an asterisk and the mail server's master user. For Dovecot, the master user is `*dovecotadmin`. In this case, you need to set `OX_IMAP_LOGIN` to `'{}*dovecotadmin'`. The OX Connector interprets the curly braces as a template for the primary email address. You can add any user attribute inside the curly braces if needed, for example, `{username}` while empty braces are for `primaryMailAddress`.

OX_FUNCTIONAL_ACCOUNT_LOGIN_TEMPLATE

A template that defines the value which is used by OX to log in to the functional account inbox. If this value is empty it is set to a concatenation of the functional account LDAP entry UUID and the user LDAP uid.

This template can include the functional account entry UUID (*fa_entry_uuid*), the functional account email address (*fa_email_address*) and any OX user UDM property (including the user's *entry_uuid* and *dn*). Every

UDM property used in this template must be enclosed by `{{ }}` e.g `{{fa_entry_uuid}}{{username}}`. Multiple values can optionally be separated by other text.

Required	Type	Initial value
No	String	N/A

Note

If the UCS *OX App Suite* is used, this app setting can be left empty, which is equivalent to using the value `{{fa_entry_uuid}}{{username}}`.

OX Connector installations that previously only used the functional account entry UUID should configure this app setting to `{{fa_entry_uuid}}`.

Some examples:

```
"{{fa_entry_uuid}}::{{entry_uuid}}" # Functional account entry UUID and␣
↪user UUID separated by two colons.
"{{username}}+{{fa_entry_uuid}}+{{dn}}" # username, functional account␣
↪entry UUID and user dn separated by a '+'
"{{fa_email_address}}*dovecotadmin" # Concatenation of functional account's␣
↪mail address and the string *\*dovecotadmin
```

Note

In cases where SSO is to be used, this variable has to be appended with an asterisk and the mail server's master user. For Dovecot this would be **dovecotadmin*. In this case `OX_FUNCTIONAL_ACCOUNT_LOGIN_TEMPLATE` can be set to `'{{fa_email_address}}*dovecotadmin'`. The resulting login value for the functional account would then look like this:

```
myfunctional_account@maildomain.de*dovecotadmin
```

OX_USER_IDENTIFIER

Defines which UDM user property is used as the unique user identifier for OX. If this app setting is not set the **OX Connector** will use the `username` property by default.

Required	Type	Initial value
No	String	N/A

Note

Only a UDM user property that contains a **single value** which is **not None** (mandatory UDM property) is a valid option. In case a UDM user property that contains an empty value or a list of values is specified, the **OX Connector** will enter an error state which needs to be resolved manually by simply setting a valid value.

OX_GROUP_IDENTIFIER

Defines which UDM group property is used as the unique group identifier for OX. If this app setting is not set the **OX Connector** will use the `name` property by default.

Required	Type	Initial value
No	String	N/A

Note

Only a UDM group property that contains a **single value** which is **not None** (mandatory UDM property) is a valid option. In case a UDM group property that contains an empty value or a list of values is specified, the **OX Connector** will enter an error state which needs to be resolved manually by simply setting a valid value.

OX_SHARED_ACCOUNT_IDENTIFIER

Defines which UDM shared account property is used as the unique shared account identifier for OX. If this app setting is not set the **OX Connector** will use the `name` property by default.

Required	Type	Initial value
No	String	N/A

Note

Only a UDM shared account property that contains a **single value** which is **not None** (mandatory UDM property) is a valid option. In case a UDM shared account property that contains an empty value or a list of values is specified, the **OX Connector** will enter an error state which needs to be resolved manually by simply setting a valid value.

OX_CONNECTOR_LOG_LEVEL

Defines the log level for the ox-connector app. If this app setting is not set the **OX Connector** will use `INFO` by default.

Required	Type	Initial value
No	String	`INFO`

OX_ENABLE_DEPUTY_PERMISSIONS

Enables the provisioning of OX deputy permissions. Administrators then can set, modify, or delete deputy permissions for users in the UMC.

For example, administrators can grant `user01` the roles *Viewer*, *Editor*, and *Author* for the calendar and mail module for `user02`. Furthermore, `user01` can send emails on behalf of `user02`.

The default value is `False`. To enable the feature, set the app setting *OX_ENABLE_DEPUTY_PERMISSIONS* (page 18) to `True`.

To see the feature in the UMC, you need to enable the UMC representation for the extended attribute either after the app installation, or after the configuration. Run the command in Listing 3.1 either on the *UCS Primary Directory Node* or a *UCS Backup Directory Node*.

Listing 3.1: Activate the UMC representation of the enabled deputy permission feature.

```
$ univention-directory-manager \
   settings/extended_attribute modify \
   --dn "cn=oxDeputyPermissionGivenTo,cn=open-xchange,cn=custom attributes,
```

(continues on next page)

(continued from previous page)

```
↪cn=univention,$(ucr get ldap/base)" \
  --set disableUDMWeb="0"
```

Required	Type	Initial value
No	Boolean	`False`

Important

The *Deputy Permissions* feature requires *OX App Suite* version >= 8.

Users can modify the deputy permissions on their own in *OX App Suite*. The provisioning in the **OX Connector** app through UCS overwrites these settings.

See also

Deputy permissions : Technical Documentation[15]
for more information about OX deputy permissions.

OX_CONNECTOR_STOP_ON_ERROR

Changes the behavior of the app **OX Connector** when it handles errors during synchronization. It can either stop on any error and retry the failed action until it succeeds or the Administrator resolves it manually (`True`). The App can also continue with other actions in its queue and instead put the failed action "aside", meaning it will no longer interfere with the execution of the connector, but can be examined by the Administrator at any later stage (`False`). See also *Troubleshooting* (page 29).

Required	Type	Initial value
No	Boolean	`True`

3.2 Univention Configuration Registry variables

ox/context/id

The app setting *DEFAULT_CONTEXT* (page 15) sets the value of the Univention Configuration Registry variable *ox/context/id* (page 19).

Upon installation of the app **OX Connector**, the OX Connector creates the extended attribute `oxContext` and uses the value from *ox/context/id* (page 19) as initial value for the extended attribute `oxContext`.

When an administrator creates a new user account that the OX Connector synchronizes, UDM sets the OX context for the user account to value of the extended attribute `oxContext`.

Caution

The UCR variable *ox/context/id* (page 19) **isn't** for manual usage.

Changing the variable **doesn't** change the OX context on existing user accounts.

Changing the value of the app setting *DEFAULT_CONTEXT* (page 15) does **neither** change *ox/context/id* (page 19) **nor** the extended attribute `oxContext`.

[15] https://documentation.open-xchange.com/8/middleware/permissions_and_capabilities/deputy_permission.html

3.3 User attribute mapping

Added in version 2.2.9: Modify the mapping between *Open-Xchange* and *UDM* properties.

Since version 2.2.9, you can modify the mapping between *Open-Xchange* and *UDM* properties using the script **`change_attribute_mapping.py`** provided with the app. The script creates a JSON file that stores information about the Open-Xchange properties and other information useful for user provisioning.

Don't modify the file manually, but only with the script. The JSON file locates at `/var/lib/univention-appcenter/apps/ox-connector/data/AttributeMapping.json`. If the file doesn't exist, the OX Connector uses the default mapping defined in `/usr/lib/python3.9/site-packages/univention/ox/provisioning/default_user_mapping.py` inside the Docker container of the app.

The script allows the following operations:

`modify`
: performs operations that change the current mapping.

`restore_default`
: restores the default mapping.

`dump`
: writes the current JSON mapping to console.

With the *modify* operation, you can use the following additional operations:

`--set`
: Changes the UDM property used for an Open-Xchange property provisioning. Listing 3.2 shows how to set the mapping of the Open Xchange property `userfield01` to the UDM property `description`.

Listing 3.2: Sets the mapping of an Open-Xchange property to an UDM property.

```
$ python3 /var/lib/univention-appcenter/apps/ox-connector/data/resources/
↪change_attribute_mapping.py \
   modify \
   --set userfield01 description
```

It's possible to use the *`--set`* (page 20) arguments multiple times in the same invocation. Listing 3.3 shows an example that sets the mapping of the Open-Xchange properties `userfield01` and `given_name` to the UDM properties `description` and `custom_attribute`.

Listing 3.3: Sets the mapping of multiple Open-Xchange properties to multiple UDM properties.

```
$ python3 /var/lib/univention-appcenter/apps/ox-connector/data/resources/
↪change_attribute_mapping.py \
   modify \
   --set userfield01 description \
   --set given_name custom_attribute
```

`--unset`
: Removes the Open-Xchange property from the mapping if it isn't marked as required. You can use it to remove properties from the synchronization.

Listing 3.4: Unset the OX property `userfield01`.

```
$ python3 /var/lib/univention-appcenter/apps/ox-connector/data/resources/
↪change_attribute_mapping.py \
   modify \
   --unset userfield01
```

--set_alternatives

Sets alternative UDM properties used for the synchronization if the main one is `None`. Listing 3.5 shows an example to set the theoretical attributes `CustomAttributeUserMail` and `CustomAttributeUserMail2` as alternatives to the Open-Xchange property `email1`.

Listing 3.5: Set theoretical attributes as alternatives to an Open-Xchange property.

```
$ python3 /var/lib/univention-appcenter/apps/ox-connector/data/resources/
↪change_attribute_mapping.py \
   modify \
   --set_alternatives email1 CustomAttributeUserMail CustomAttributeUserMail2
```

--unset_alternatives

Unset the current alternatives for an OX property

Listing 3.6: Unset the alternative attributes to the OX property `email1`.

```
$ python3 /var/lib/univention-appcenter/apps/ox-connector/data/resources/
↪change_attribute_mapping.py \
   modify \
   --unset_alternatives email1
```

If you previously used the attribute mapping feature of the *OX App Suite* app from the App Center, you can migrate it by running the following command on the UCS system where you installed the OX App Suite. You then use the output of the script as command and run it on the UCS system where the OX Connector is running.

```
python3 <<EOF
  from univention.config_registry import ConfigRegistry
  ucr = ConfigRegistry()
  ucr.load()

  changed_mapping_single = {
    'displayname': 'display_name',
    'givenmame': 'given_name',
    'surname': 'sur_name',
    'categories': 'employee_type',
    'quota': 'max_quota',
    }

  changed_mapping_multi = {
    'telephone_business': ['telephone_business1', 'telephone_business2'],
    'telephone_home': ['telephone_home1', 'telephone_home2'],
  }


  ucr_ldap2ox = ucr.get('ox/listener/user/ldap/attributes/mapping/ldap2ox', '').
↪strip()
  ucr_ldap2oxmulti = ucr.get('ox/listener/user/ldap/attributes/mapping/ldap2oxmulti
↪', '').strip()
  command = []
  if ucr_ldap2ox:
    for entry in ucr_ldap2ox.split():
      value, key = entry.split(':', 1)
      if value is None:
        command.append(f"--unset {changed_mapping_single.get(key, key)}")
      else:
        command.append(f"--set {changed_mapping_single.get(key, key)} {value}")
```

(continues on next page)

(continued from previous page)

```
  if ucr_ldap2oxmulti:
    ldap2oxmulti = {}
    for entry in ucr_ldap2oxmulti.split():
      value, key = entry.split(':', 1)
      if value is None:
        for v in changed_mapping_multi.get(key, [key]):
          command.append(f"--unset {v}")
      else:
        for v in changed_mapping_multi.get(key, [key]):
          command.append(f"--set {v} {value}")
  if command:
    print("Run the following command on the ox-connector server to update␣
↪attribute mapping:")
    print("python3 /var/lib/univention-appcenter/apps/ox-connector/data/resources/
↪change_attribute_mapping.py modify " + " ".join(command))
  else:
    print("Nothing to do.")
EOF
```

3.4 Import additional CA certificates

Added in version 2.3.0: Allow the connector to import additional CA certificates

The **OX Connector** app in UCS runs as a container with its own CA certificate store. By default, the app imports the UCS root CA certificate into the CA store to enable a secure connection to the UCS LDAP directory. You may need additional CA certificates for the **OX Connector** app, for example, when provisioning to a remote *OX App Suite* installation.

To add certificates to the certificate store in the **OX Connector**, use the following steps on the system where the app is installed:

1. Create the `/var/lib/univention-appcenter/apps/ox-connector/data/conf/ca-certificates/` directory.
2. Copy the CA certificate files in PEM format with the ending `.pem` into this directory. Listing 3.7 shows an example.

Listing 3.7: Examples for additional CA certificates

```
$ file /var/lib/univention-appcenter/apps/ox-connector/data/conf/ca-
↪certificates/*.pem
.../ox-connector/data/conf/ca-certificates/cert1.pem: PEM certificate
.../ox-connector/data/conf/ca-certificates/cert2.pem: PEM certificate
```

3. Manually reconfigure the OX Connector with the command in Listing 3.8. The **OX Connector** app automatically adds the certificates to its certificate store, also during app updates.

Listing 3.8: Manually reconfigure the OX Connector

```
$ univention-app configure ox-connector
```

CHAPTER

FOUR

ARCHITECTURE

The **OX Connector** app architecture consists of the following elements:

- The operating environment UCS with the App Center and the Docker engine running OX Connector.
- The OX Connector software inside a Docker image.
- The OpenLDAP LDAP directory in UCS as identity management source for OX App Suite.

4.1 Overview

The **OX Connector** app consists of a Docker image with all the software needed to provision user identities from UCS identity management to OX App Suite. The OX connector connects to the OX App Suite SOAP API and creates, updates, or deletes object entries in OX App Suite depending on what changed in the UCS LDAP directory with relevance to OX App Suite.

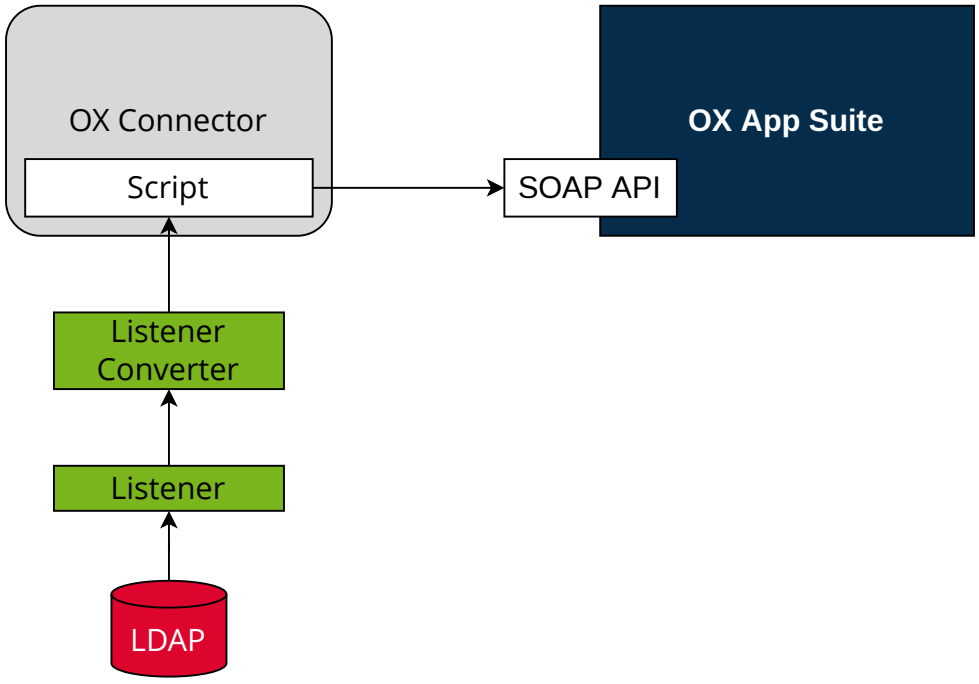

Fig. 4.1: OX Connector app architecture
View focuses on the elements LDAP Directory, Listener, Listener Converter, OX Connector with the provisioning script, OX App Suite, and its SOAP API.

LDAP
: The OpenLDAP software provides the *LDAP* directory in UCS. The LDAP directory stores all identity and infrastructure data of the UCS domain. For more information, see LDAP directory[16] in *UCS 5.2 Manual* [2].

[16] https://docs.software-univention.de/manual/5.2/en/domain-ldap/ldap-directory.html#domain-ldap

Listener
The App Center creates a *Listener* module for the **`OX Connector`** app, when it installs the app on a UCS system. The *Listener* writes the `UniventionObjectIdentifier` of the LDAP object that changed, in JSON format to `/var/lib/univention-appcenter/listener/ox-connector/`*`timestamp`*`.json`. Each change creates one file.

Listener Converter
The *Listener Converter* is a services running on UCS with the following responsibility:

1. Process the JSON files from the *Listener* ordered by the timestamp in the filename.
2. Request the LDAP object attributes through UDM for each `UniventionObjectIdentifier`.

The converter writes the results in JSON format to `/var/lib/univention-appcenter/apps/ox-connector/data/listener/`*`timestamp`*`.json`.

OX Connector
OX Connector connects the UCS identity management with OX App Suite. The connector receives data about changes in the LDAP directory. A *Script* handles the data, processes it and sends it to the *SOAP API* in OX App Suite.

Script
The *Script* runs inside the Docker container of the OX Connector. It handles the files in JSON format from the *Listener Converter*, consumes it and sends data to the *SOAP API*.

The *Script* doesn't run multiple times at the same time.

OX App Suite
OX App Suite is the groupware and collaboration software from Open-Xchange.

SOAP API
OX App Suite uses SOAP[17] as network protocol to receive data and run remote procedure calls. The connector uses the SOAP API to create, update, or delete object entries in OX App Suite.

4.2 How the connector works

The OX Connector reacts on changes in the LDAP directory in UCS and relies on modules in the Univention Directory Manager (UDM) modules. UDM is a layer on top of the LDAP directory in UCS.

UCS provides the following UDM modules:

- `users/user`
- `groups/group`

The OX Connector provides the following UDM modules:

- `oxmail/oxcontext`
- `oxresources/oxresources`
- `oxmail/accessprofile`

The OX Connector reacts on changes to the listed UDM modules and sends data to the SOAP API in OX App Suite.

4.2.1 Access profiles

Upon changes in the UDM module `oxmail/accessprofile`, the connector rewrites the local file `/var/lib/univention-appcenter/apps/ox-connector/data/ModuleAccessDefinitions.properties` and doesn't send data to the SOAP API in OX App Suite. The module handles the user rights and roles in OX App Suite. Administrators find the *access profiles* in UMC in the module LDAP directory at *open-xchange* ‣ *accessprofile*.

[17] https://en.wikipedia.org/wiki/SOAP

4.2.2 Provisioning

In detail, the provisioning has the following steps, see Fig. 4.2:

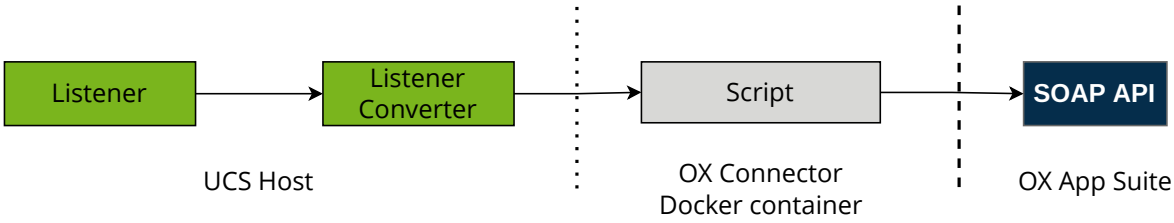

Fig. 4.2: Provisioning procedure

1. The *Listener* writes one file per change.
2. The *Listener Converter* writes one file per change with the LDAP object attributes.
3. The *Listener Converter* triggers the *Script* in the OX Connector Docker container.
4. In the Docker container, the *Script* consumes all the JSON files from the *Listener Converter*.
5. After the *SOAP API* received the data and processed them successfully, the *Script* deletes each JSON file.
6. The *Listener Converter* waits for 5 seconds and restarts at step 2.

For more information about the file contents of the *Listener* and *Listener Converter*, see *Overview* (page 23).

4.3 Provisioned attributes

The **`OX Connector`** provisions a lot of attributes to OX App Suite. A detailed description is beyond the scope of this document.

The OX Connector comes with the source code. The user attributes for provisioning locate in the function `update_user()` in `univention-ox-provisioning/univention/ox/provisioning/users.py` inside the Docker container. To view the attributes, for example with **`vim`**, run the following command on the UCS system with OX Connector installed. Replace `$version` with the proper Python version used in the Connector:

Listing 4.1: Example for how to view the definition of provisioned attributes.

```
$ univention-app shell ox-connector \
  cat /usr/lib/python"$version"/site-packages/univention/ox/provisioning/users.py \
  | vim -
```

Likewise, the attributes for groups, context, and resources locate in the respective source files in the `update_*()` function.

4.4 Database of old entries

Added in version 3.0.0.

OX App Suite creates an *internal ID* for every user object it creates or updates. The OX Connector saves this *internal ID* in its own database, when it processed the objects without errors. The connector doesn't store that ID in the UCS LDAP directory, but maintains a database in which it stores the data old object's it processed for later reference (i.e., for retrieving the *internal ID*)

The database files is located at `/var/lib/univention-appcenter/apps/ox-connector/data/listener/ox-connector.db`. The table is named *old*. Administrators are highly advised to use the CLI the App provides to manipulate this database, see *CLI to monitor the current state* (page 29).

When the *Listener Converter* updates groups in OX App Suite, the request to the *SOAP API* must include the internal ID of all group members. The connector would need to ask the database of OX App Suite for the *internal ID* of each group member, involving network requests and database queries. To speed up the processing, the OX Connector uses the *internal ID* from the database.

CHAPTER

FIVE

LIMITATIONS

To ensure a smooth operation of the **OX Connector** app on UCS, you as administrator need to know the following limitations.

5.1 Integration of OX Connector and OX App Suite app

Starting with version 2.1.2, Univention supports the use of the **OX Connector** for the **OX App Suite** app from Univention App Center. The **OX Connector** handles the provisioning, while the **OX App Suite** delivers the actual groupware.

However, the OX Connector needs administrative credentials to create context objects in the database for the **OX App Suite**. The installation process doesn't know these credentials. Therefore, you need to verify the configuration of the **OX Connector** after you have successfully installed **OX App Suite**. The reconfiguration runs automatically if, and only if, both apps locate on the same UCS system.

If not, you find the password in the file `/etc/ox-secrets/master.secret` on the UCS system running **OX App Suite**. The username of the administrative account is `oxadminmaster`. You need to set the credentials in the app settings of the **OX Connecor**, see *Configuration* (page 15).

5.2 How the Connector handles faulty items

The **OX Connector** knows two strategies how to handle faulty items it can't synchronize. You can choose which strategy to use: *App Settings* (page 15).

5.2.1 OX Connector continues after faulty items

When the **OX Connector** encounters a faulty queue item that it can't process, it continues with the next queue items. The OX Connector puts the faulty item aside for the Administrator to examine at a later stage. The problem is written in the log file, see *Log files* (page 29).

The app ships a CLI to manage the list of errors, see *CLI to monitor the current state* (page 29).

As administrator, you need to monitor the list of errors manually and decide what to do (delete or retry). Meanwhile, the **OX Connector** continues to process data it gets from the *Listener*.

Note that certain errors are excluded from that behavior. When the **OX Connector** encounters a problem that hints to a network error, it retries this one task over and over again as continuing will most probably result in the same error for all items anyway. Synchronizing objects from the UDM module `oxmail/oxcontext` will also be retried as these objects are extremely important to be in sync. All following items in the queue will likely fail, therefore the app does not just continue in this case. The strategy of stopping instead of continuing is also described in the next chapter *OX Connector stops at faulty items* (page 27).

5.2.2 OX Connector stops at faulty items

When the **OX Connector** encounters a faulty queue item that it can't process, it stops the provisioning at the item and the problematic task in the *Listener Converter* log file, see *Log files* (page 29).

Despite the stop, the *Listener* continues to add items to the queue. After the administrator removed the faulty queue item, the Listener Converter continues to process the queue and also takes care of the added items.

As administrator, you need to resolve that conflict manually when it happens, see *Provisioning stops working* (page 31). After the conflict resolution, the connector continues to process the provisioning queue.

5.3 No plausibility validation in access profile rights

The **`OX Connector`** app doesn't evaluate permission level for created *access profiles* and tries to create any access profile.

For more information, see OX App Suite Permission Level[18].

[18] https://oxpedia.org/wiki/index.php?title=AppSuite:Permission_Level

CHAPTER

SIX

TROUBLESHOOTING

When you encounter problems with the operation of the **OX Connector** app, this section provides information where you can look closer into and to get an impression about what's going wrong.

6.1 Log files

The **OX Connector** app produces different logging information in different places.

Listener Converter: `/var/log/univention/listener_modules/ox-connector.log`
Contains log information from the *Listener Converter* about create, update and delete actions of objects.

It also shows warnings and errors when the OX Connector configuration isn't correct, or the connector can't establish a connection to the *SOAP API*.

Database management script: `/var/lib/univention-appcenter/apps/ox-connector/data/db.log`
Contains log information from the *Database management script* that is described below.

App Center: `/var/log/univention/appcenter.log`
Contains log information around activities in the App Center.

The App Center writes OX Connector relevant information to this file, when you run app lifecycle tasks like install, update and uninstall or when you change the app settings.

Domain join: `/var/log/univention/join.log`
Contains log information from the join processes. When the App Center install OX Connector, the app also joins the domain.

6.2 Checking the Listener

Before checking the OX Connector, you may want to have a look at the connection between the *Listener* and the *Listener Converter*. The Listener should create files and the Listener Converter should translate these files rather quickly.

Listing 6.1: Verify the number of unprocessed files for the *Listener*.

```
$ DIR_LISTENER="/var/lib/univention-appcenter/listener/ox-connector"
$ ls -1 "$DIR_LISTENER"/*.json 2> /dev/null | wc -l
0
```

If files here are not created upon a change or are piling up, this indicates a problem in the Listener or Listener Converter. See *Log files* (page 29).

6.3 CLI to monitor the current state

The OX Connector ships a command-line interface that you can use to query and manipulate the database it uses to keep track of current tasks, objects already synced and errors it may have found.

Listing 6.2: List all commands of the CLI.

```
$ /usr/sbin/univention-ox-connector-task-management --help
```

The tool operates on the **SQLite** database `/var/lib/univention-appcenter/apps/ox-connector/data/listener/ox-connector.db`. The terminology of the tool is as follows:

Tasks
: A database table managed by the OX Connector. A row represents an active task. The OX Connector iterates over all tasks and synchronizes them to the OX App Suite.

Old
: A database table managed by the OX Connector. A row represents the state of an item at the moment it was successfully synchronized. It is more or less a copy of a former task. Needed when certain items are synchronized and reference other items (e.g., when synchronizing a group that contains users). Also used for faster look-ups by storing the database ID given by OX.

Morgue
: A database table managed by the OX Connector. A row represents a failed task. It was moved automatically or manually to this table and is not actively processed by the OX Connector. Administrators can examine the items in the morgue and decide how to proceed with them (see below).

6.3.1 Health check

First, have a look at the log file for the *Listener Converter* and look for warnings and errors, see *Log files* (page 29).

Second you can get a brief summary of current tasks. This can indicate if the OX Connector can process the items fast enough or at all.

Listing 6.3: Show all tasks the OX Connector is yet to process.

```
$ /usr/sbin/univention-ox-connector-task-management summarize-tasks
$ /usr/sbin/univention-ox-connector-task-management search-tasks
```

Third, you can get a brief summary of past errors. Every item is an object not synchronized. Note that this only makes sense should you have chosen *OX Connector continues after faulty items* (page 27).

Listing 6.4: Show all items in the morgue.

```
$ /usr/sbin/univention-ox-connector-task-management search-morgue
```

6.3.2 Handling errors

You can decide what to do with the items that have been moved to the morgue. All commands assume that you have the `UniventionObjectIdentifier` of that object. For each item you have the option to

1. Delete it from the list: It is as if this item never hit the OX Connector. The underlying object can of course be synchronized again if it is modified in the LDAP directory (creating a completely new item in the OX Connector's tasks).

   Listing 6.5: Remove an item from the morgue.

   ```
   $ /usr/sbin/univention-ox-connector-task-management remove-from-morgue --obj-
   ↪id=...
   ```

2. Retry the very same item: The erroneous item in the list is again copied to the list of tasks, assuming that the problem is now fixed (e.g., a validation on the OX App Suite's side has been disabled).

Listing 6.6: Retry an item from the morgue.

```
$ /usr/sbin/univention-ox-connector-task-management retry-from-morgue --obj-
↪id=...
```

3. Fresh synchronization of the object: The object is again put into the list of tasks but not with the attributes it had when the synchronization happened (and failed). Instead, it is freshly fetched from the LDAP database. This only works for the first object found, so asterisks may not do what you expect.

Listing 6.7: Re-sync an existing item via UDM.

```
$ /usr/sbin/univention-ox-connector-task-management resync-item --obj-id=...
```

6.4 Provisioning stops working

When the provisioning stopped working, a previous change in UDM is a probable reason and the OX Connector doesn't know how to proceed. The connector retries the action over and over again until an administrator repairs the cause manually.

First, see the *Log files* (page 29) and look for warnings and errors. If it's not a temporary problem like for example network connectivity, the fix requires manual action.

As a last resort, the administrator can move the task aside. The log file reveals the `Database ID` of that object (e.g. `uid=...; $object_identifier; tasks:$database_id`).

Listing 6.8: Retry an error from the list.

```
$ /usr/sbin/univention-ox-connector-task-management move-task-to-morgue --task-id=
↪$database_id  --error-msg="Manual intervention after careful consideration"
```

6.4.1 Re-provision all data

Warning

Depending on the number of users and groups in the UCS LDAP directory, this task may take a lot of time.

Reprovisioning all data isn't recommended.

The following command reads all UDM objects from the UCS LDAP directory and adds them to the provisioning queue:

Listing 6.9: Re-provisioning all UDM objects to OX App Suite

```
$ univention-directory-listener-ctrl resync ox-connector
```

The re-provisioning won't run any *delete* operations, because the Listener only adds existing UDM objects to the queue.

Caution

The OX Connector may decide to delete objects based on data in the JSON files. For example `isOxGroup = Not` in a group object.

6.5 Ensuring the OX database ID integrity

The *internal ID* of objects in the database of OX App Suite can become corrupted, for example after a backup restore of the database. For more information about the cache, see *Database of old entries* (page 25).

To rewrite that cache, run the following commands:

Listing 6.10: Rebuild cache for *internal ID*

```
$ /usr/sbin/univention-ox-connector-task-management rewrite-ox-db-id
```

Tip

Retrieve all users per context in one request
Rebuilding the cache may take a long time and depends on the amount of users in the OX App Suite database.

`/usr/sbin/univention-ox-connector-task-management rewrite-ox-db-id --build-cache-size=1000` can speed up the rebuild, because it retrieves up to 1000 users of one context with one request.

Warning

Memory consumption
On the UCS system with the OX Connector, the rebuild process may use up to 1 GB memory per 10,000 users in the database for OX App Suite.

System load
Furthermore, the process may generate a lot of load on the OX App Suite system and the OX Connector app.

6.6 Duplicated *displaynames*

In OX Connector version 2.2.0 the UDM property *oxDisplayName* does not have a unique constraint anymore.

If duplicate values are used, but OX is not prepared for that, the *SOAP API* calls will fail with the following exception.

```
2023-05-30 11:59:31 WARNING Traceback (most recent call last):
2023-05-30 11:59:31 WARNING   File "/tmp/univention-ox-connector.listener_trigger",
↪ line 324, in run_on_files
2023-05-30 11:59:31 WARNING     f(obj)
2023-05-30 11:59:31 WARNING   File "/usr/lib/python3.9/site-packages/univention/ox/
↪provisioning/__init__.py", line 86, in run
2023-05-30 11:59:31 WARNING     modify_user(obj)
2023-05-30 11:59:31 WARNING   File "/usr/lib/python3.9/site-packages/univention/ox/
↪provisioning/users.py", line 420, in modify_user
2023-05-30 11:59:31 WARNING     user.modify()
2023-05-30 11:59:31 WARNING   File "/usr/lib/python3.9/site-packages/univention/ox/
↪soap/backend.py", line 477, in modify
2023-05-30 11:59:31 WARNING     super(SoapUser, self).modify()
2023-05-30 11:59:31 WARNING   File "/usr/lib/python3.9/site-packages/univention/ox/
↪soap/backend.py", line 180, in modify
2023-05-30 11:59:31 WARNING     self.service(self.context_id).change(obj)
2023-05-30 11:59:31 WARNING   File "/usr/lib/python3.9/site-packages/univention/ox/
↪soap/services.py", line 536, in change
2023-05-30 11:59:31 WARNING     return self._call_ox('change', usrdata=user)
```

(continues on next page)

(continued from previous page)

```
2023-05-30 11:59:31 WARNING   File "/usr/lib/python3.9/site-packages/univention/ox/
↪soap/services.py", line 163, in _call_ox
2023-05-30 11:59:31 WARNING     return getattr(service, func)(**kwargs)
2023-05-30 11:59:31 WARNING   File "/usr/lib/python3.9/site-packages/zeep/proxy.py
↪", line 46, in __call__
2023-05-30 11:59:31 WARNING     return self._proxy._binding.send(
2023-05-30 11:59:31 WARNING   File "/usr/lib/python3.9/site-packages/zeep/wsdl/
↪bindings/soap.py", line 135, in send
2023-05-30 11:59:31 WARNING     return self.process_reply(client, operation_obj,␣
↪response)
2023-05-30 11:59:31 WARNING   File "/usr/lib/python3.9/site-packages/zeep/wsdl/
↪bindings/soap.py", line 229, in process_reply
2023-05-30 11:59:31 WARNING     return self.process_error(doc, operation)
2023-05-30 11:59:31 WARNING   File "/usr/lib/python3.9/site-packages/zeep/wsdl/
↪bindings/soap.py", line 329, in process_error
2023-05-30 11:59:31 WARNING     raise Fault(
2023-05-30 11:59:31 WARNING zeep.exceptions.Fault: The displayname is already used;
↪ exceptionId 1170523631-4
```

To fix this issue, a change in the *OX App Suite* configuration is required. Add the following lines to the `user.properties` file.

```
com.openexchange.user.enforceUniqueDisplayName=false
com.openexchange.folderstorage.database.preferDisplayName=false
```

Note

This is configured by default in the *OX App Suite* installation from the App center.

6.7 Traceback provisioning *groups*

When an ox group is synchronized, the **`OX Connector`** obtains information about all its users by reading from the *listener/old* directory where the latest version of the objects that have already been synchronized is stored. If any user is part of such group but is not in *listener/old*, the **`OX Connector`** will fail with a traceback like the following:

```
2024-11-15 16:06:33 INFO    Group oxgroup will be OX Group
2024-11-15 16:06:33 INFO    Error while processing /var/lib/univention-appcenter/
↪apps/ox-connector/data/listener/2024-11-15-15-51-31-669745.json
2024-11-15 16:06:33 INFO    This is consecutive error #11
2024-11-15 16:06:33 INFO    Sleeping for 0 sec
2024-11-15 16:06:33 WARNING Traceback (most recent call last):
2024-11-15 16:06:33 INFO    Successfully processed 0 files during this run
2024-11-15 16:06:33 WARNING   File "/tmp/univention-ox-connector.listener_trigger",
↪ line 419, in run_on_files
2024-11-15 16:06:33 WARNING     function(obj)
2024-11-15 16:06:33 WARNING   File "/usr/lib/python3.9/site-packages/univention/ox/
↪provisioning/__init__.py", line 115, in run
2024-11-15 16:06:33 WARNING     for new_obj in get_group_objs(obj):
2024-11-15 16:06:33 WARNING   File "/usr/lib/python3.9/site-packages/univention/ox/
↪provisioning/__init__.py", line 173, in get_group_objs
2024-11-15 16:06:33 WARNING     user_obj = univention.ox.provisioning.helpers.get_
↪old_obj(user)
2024-11-15 16:06:33 WARNING   File "/tmp/univention-ox-connector.listener_trigger",
↪ line 76, in _get_old_object
```

(continues on next page)

(continued from previous page)

```
2024-11-15 16:06:33 WARNING     raise Exception(f"Old object file {path_to_old_
↪user} for {distinguished_name} does not exist!\nYou need to re-provision \"
↪{distinguished_name}\" (see https://docs.software-univention.de/ox-connector-app/
↪latest/troubleshooting  .html#traceback-provisioning-groups).")
2024-11-15 16:06:33 WARNING Exception: Old object file /var/lib/univention-
↪appcenter/apps/ox-connector/data/listener/old/d10338de-3144-103f-8ea2-
↪f39fa7a811dd.json for uid=oxuser1,cn=users,dc=example,dc=com does not exist!
2024-11-15 16:06:33 WARNING You need to re-provision "uid=oxuser1,cn=users,
↪dc=example,dc=com" (see https://docs.software-univention.de/ox-connector-app/
↪latest/troubleshooting.html#traceback-provisioning-groups).
```

You need to re-provision the user object (*uid=oxuser1,cn=users,dc=example,dc=com* in this case) manually. Follow the instructions in *Handling errors* (page 30) to synchronize the missing users. After this manual intervention the connector automatically continues with the synchronization of the group object.

6.8 Collect information for support ticket

Before you open a support ticket, make sure to collect and provide relevant details about your case, so that the Univention Support team can help you:

- Provide relevant details about your environment[19].
- Provide the relevant messages and tracebacks from *Log files* (page 29), specifically the *Listener Converter*.
- Describe the steps that can reproduce the faulty behavior.
- Describe the expected behavior.
- Provide data from the provisioning that causes the error.

6.9 Invalid values for OX_USER_IDENTIFIER or OX_GROUP_IDENTIFIER

Only a UDM user property (or UDM group property in case of OX_GROUP_IDENTIFIER) that contains a **single value** which is **not None** is a valid option. In case a UDM property that contains an empty value or a list of values is specified, the **`OX Connector`** will enter an error state which needs to be resolved manually by simply setting a valid value.

Setting invalid values for the app settings *OX_USER_IDENTIFIER* or *OX_GROUP_IDENTIFIER* will lead to the following errors:

```
2024-01-11 13:57:39 WARNING Traceback (most recent call last):
2024-01-11 13:57:39 WARNING   File "/tmp/univention-ox-connector.listener_trigger",
↪ line 351, in run_on_files
2024-01-11 13:57:39 WARNING     function(obj)
2024-01-11 13:57:39 WARNING   File "/usr/lib/python3.9/site-packages/univention/ox/
↪provisioning/__init__.py", line 86, in run
2024-01-11 13:57:39 WARNING     modify_user(obj)
2024-01-11 13:57:39 WARNING   File "/usr/lib/python3.9/site-packages/univention/ox/
↪provisioning/users.py", line 454, in modify_user
2024-01-11 13:57:39 WARNING     user.modify()
2024-01-11 13:57:39 WARNING   File "/usr/lib/python3.9/site-packages/univention/ox/
↪soap/backend.py", line 475, in modify
2024-01-11 13:57:39 WARNING     super(SoapUser, self).modify()
2024-01-11 13:57:39 WARNING   File "/usr/lib/python3.9/site-packages/univention/ox/
↪soap/backend.py", line 176, in modify
```

(continues on next page)

[19] https://help.univention.com/faq#posting-guidelines

(continued from previous page)

```
2024-01-11 13:57:39 WARNING     assert self.name is not None
2024-01-11 13:57:39 WARNING No name for this attribute. Missing or misconfigured␣
↪identifier app settings
2024-01-11 13:57:39 WARNING (OX_USER_IDENTIFIER or OX_GROUP_IDENTIFIER) might be␣
↪the reason, see
2024-01-11 13:57:39 WARNING https://docs.software-univention.de/ox-connector-app/
↪latest/troubleshooting.html#invalid-values-for-ox-user-identifier-or-ox-group-
↪identifier
2024-01-11 13:57:39 WARNING for more information.
```

```
setting "users" udm property for groups
2024-01-11 13:59:36 WARNING Traceback (most recent call last):
2024-01-11 13:59:36 WARNING   File "/tmp/univention-ox-connector.listener_trigger",
↪ line 351, in run_on_files
2024-01-11 13:59:36 WARNING     function(obj)
2024-01-11 13:59:36 WARNING   File "/usr/lib/python3.9/site-packages/univention/ox/
↪provisioning/__init__.py", line 108, in run
2024-01-11 13:59:36 WARNING     modify_group(new_obj)
2024-01-11 13:59:36 WARNING   File "/usr/lib/python3.9/site-packages/univention/ox/
↪provisioning/groups.py", line 146, in modify_group
2024-01-11 13:59:36 WARNING     group.modify()
2024-01-11 13:59:36 WARNING   File "/usr/lib/python3.9/site-packages/univention/ox/
↪soap/backend.py", line 180, in modify
2024-01-11 13:59:36 WARNING     self.service(self.context_id).change(obj)
2024-01-11 13:59:36 WARNING   File "/usr/lib/python3.9/site-packages/univention/ox/
↪soap/services.py", line 607, in change
2024-01-11 13:59:36 WARNING     return self._call_ox('change', grp=grp)
2024-01-11 13:59:36 WARNING   File "/usr/lib/python3.9/site-packages/univention/ox/
↪soap/services.py", line 194, in _call_ox
2024-01-11 13:59:36 WARNING     return getattr(service, func)(**kwargs)
2024-01-11 13:59:36 WARNING   File "/usr/lib/python3.9/site-packages/zeep/proxy.py
↪", line 46, in __call__
2024-01-11 13:59:36 WARNING     return self._proxy._binding.send(
2024-01-11 13:59:36 WARNING   File "/usr/lib/python3.9/site-packages/zeep/wsdl/
↪bindings/soap.py", line 123, in send
2024-01-11 13:59:36 WARNING     envelope, http_headers = self._create(
2024-01-11 13:59:36 WARNING   File "/usr/lib/python3.9/site-packages/zeep/wsdl/
↪bindings/soap.py", line 73, in _create
2024-01-11 13:59:36 WARNING     serialized = operation_obj.create(*args, **kwargs)
2024-01-11 13:59:36 WARNING   File "/usr/lib/python3.9/site-packages/zeep/wsdl/
↪definitions.py", line 224, in create
2024-01-11 13:59:36 WARNING     return self.input.serialize(*args, **kwargs)
2024-01-11 13:59:36 WARNING   File "/usr/lib/python3.9/site-packages/zeep/wsdl/
↪messages/soap.py", line 79, in serialize
2024-01-11 13:59:36 WARNING     self.body.render(body, body_value)
2024-01-11 13:59:36 WARNING   File "/usr/lib/python3.9/site-packages/zeep/xsd/
↪elements/element.py", line 232, in render
2024-01-11 13:59:36 WARNING     self._render_value_item(parent, value, render_path)
2024-01-11 13:59:36 WARNING   File "/usr/lib/python3.9/site-packages/zeep/xsd/
↪elements/element.py", line 256, in _render_value_item
2024-01-11 13:59:36 WARNING     return self.type.render(node, value, None, render_
↪path)
2024-01-11 13:59:36 WARNING   File "/usr/lib/python3.9/site-packages/zeep/xsd/
↪types/complex.py", line 307, in render
2024-01-11 13:59:36 WARNING     element.render(node, element_value, child_path)
2024-01-11 13:59:36 WARNING   File "/usr/lib/python3.9/site-packages/zeep/xsd/
```

(continues on next page)

(continued from previous page)

```
↪elements/indicators.py", line 256, in render
2024-01-11 13:59:36 WARNING     element.render(parent, element_value, child_path)
2024-01-11 13:59:36 WARNING   File "/usr/lib/python3.9/site-packages/zeep/xsd/
↪elements/element.py", line 232, in render
2024-01-11 13:59:36 WARNING     self._render_value_item(parent, value, render_path)
2024-01-11 13:59:36 WARNING   File "/usr/lib/python3.9/site-packages/zeep/xsd/
↪elements/element.py", line 255, in _render_value_item
2024-01-11 13:59:36 WARNING     return value._xsd_type.render(node, value, xsd_
↪type, render_path)
2024-01-11 13:59:36 WARNING   File "/usr/lib/python3.9/site-packages/zeep/xsd/
↪types/complex.py", line 307, in render
2024-01-11 13:59:36 WARNING     element.render(node, element_value, child_path)
2024-01-11 13:59:36 WARNING   File "/usr/lib/python3.9/site-packages/zeep/xsd/
↪elements/indicators.py", line 256, in render
2024-01-11 13:59:36 WARNING     element.render(parent, element_value, child_path)
2024-01-11 13:59:36 WARNING   File "/usr/lib/python3.9/site-packages/zeep/xsd/
↪elements/element.py", line 232, in render
2024-01-11 13:59:36 WARNING     self._render_value_item(parent, value, render_path)
2024-01-11 13:59:36 WARNING   File "/usr/lib/python3.9/site-packages/zeep/xsd/
↪elements/element.py", line 256, in _render_value_item
2024-01-11 13:59:36 WARNING     return self.type.render(node, value, None, render_
↪path)
2024-01-11 13:59:36 WARNING   File "/usr/lib/python3.9/site-packages/zeep/xsd/
↪types/simple.py", line 96, in render
2024-01-11 13:59:36 WARNING     node.text = value if isinstance(value, etree.
↪CDATA) else self.xmlvalue(value)
2024-01-11 13:59:36 WARNING   File "/usr/lib/python3.9/site-packages/zeep/xsd/
↪types/builtins.py", line 27, in _wrapper
2024-01-11 13:59:36 WARNING     raise ValueError(
2024-01-11 13:59:36 WARNING ValueError: The String type doesn't accept collections␣
↪as value
```

6.10 Troubleshooting migration of functional accounts to shared accounts

During the *migration of functional accounts to shared accounts* (page 11), a network failure or another unexpected error can leave a shared account half-configured. You might encounter one of the following states:

Functional account still exists
: The functional account is still present, and the shared account is partially configured. Rerun the script with the same parameters as before to retry the migration.

Functional account doesn't exist anymore
: The functional account doesn't exist anymore, so the migration is nearly complete. The remaining step is to modify the email address of the shared account and remove the `tmp_` prefix. To remove the prefix, use either the *Management UI* or the **udm** command.

Management UI

Use the following steps:

1. Sign in to the *Management UI* and navigate to the LDAP directory module[20].
2. Select the container for the shared accounts. The default container is `cn=shared_accounts, cn=open-xchange,<ldap_base>`.

[20] https://docs.software-univention.de/nubus-manual/1.x/en/management/domain/ldap.html#nubus-domain-ldap

3. Open the affected shared account.
4. Change the email address and remove the `tmp_` prefix.
5. Click *Save*.

Verify that the shared account uses the expected email address and no longer has the `tmp_` prefix.

UDM command-line

To remove the prefix by using the **udm** command in Nubus for UCS, run the command shown in Listing 6.11. Define the following parameters:

SHARED_ACCOUNT
: The DN of the affected shared account, for example `"cn=test,cn=shared_accounts,cn=open-xchange,$(ucr get ldap/base)"`

EMAIL
: The email address of the shared account.

Listing 6.11: Remove the `tmp_` prefix from the email address of a shared account

```
$ export SHARED_ACCOUNT="<DN of affected shared account>"
$ export EMAIL="<email address of the shared account>"
$ udm \
   oxmail/shared_account \
   modify \
   --dn "$SHARED_ACCOUNT" \
   --set mailPrimaryAddress="$EMAIL"
```

Verify that the shared account uses the expected email address and no longer has the `tmp_` prefix.

CHAPTER

SEVEN

CHANGELOG

This changelog documents all notable changes to the OX Connector app. Keep a Changelog[21] is the format and this project adheres to Semantic Versioning[22].

7.1 v3.2.2

Released: 2026-06-25

7.1.1 Added

The new setting *OX_SHARED_ACCOUNT_IDENTIFIER* can be used to adjust the name of an *OX Shared Account*.

7.1.2 Fixed

A safeguard when finding existing OX users before their actual creation was broken in some cases. It always used the username of the UDM object for the search and did not take the configured *OX_USER_IDENTIFIER* into account.

7.2 v3.2.1

Released: 2026-06-10

7.2.1 Added

Migration script for *OX Functional Accounts* into *OX Shared Accounts*. You can now migrate your old functional account to the new shared account using the provided script. For more information, see *Migration from functional accounts to shared accounts* (page 11).

7.3 v3.2.0

Released: 2026-05-22

7.3.1 Added

Support for *OX Shared Accounts*. You can now add UDM objects for shared account and corresponding shared account permissions. For more information, see *Shared accounts* (page 10).

7.4 v3.1.0

Released: 2026-03-17

[21] https://keepachangelog.com/en/1.0.0/
[22] https://semver.org/spec/v2.0.0.html

7.4.1 Removed

The property *groups* in the Functional Accounts module has been removed. Adding groups to Functional Accounts was never supported and thus not an available option in the Univention Management Console. The logic that still made it possible to set groups via direct Univention Directory Manager access has been removed.

Important

If groups were used as Functional Account members, please add their members directly to the Functional Account. Attempting to add a group to a Functional Account will raise an error in UDM after this update.

7.5 v3.0.2

Released: 2026-03-12

7.5.1 Fixed

Users can give deputy permissions to other users (*oxDeputyPermissionGivenTo*). These references have to be updated on changes to that user. This was not done in case of a move operation in LDAP. This has been fixed.

7.6 v3.0.1

Released: 2025-10-06

7.6.1 Changed

The sub-command **`/usr/sbin/univention-ox-connector-task-management resync-item`** can now re-sync from the morgue and from the old table.

In case of consecutive errors, the OX Connector now sleeps longer and longer between runs. This is done to prevent log files filling up with the same error rather quickly. The delay between runs increases with every consecutive error up to 20 minutes.

7.6.2 Added

Added the sub-command **`/usr/sbin/univention-ox-connector-task-management rewrite-ox-db-id`**.

Added the sub-command **`/usr/sbin/univention-ox-connector-task-management export-old`**.

7.7 v3.0.0

Released: 2025-09-29

7.7.1 Changed

The App now stores its queue of tasks in a *SQLite* database instead of having all data in JSON files. This includes the tasks as well as the data of those objects already seen. GDBM based key value stores have been removed. For now, some helper scripts will not work anymore, namely *rebuild-old.db*, *remove-from-ox-db-cache*, *check_sync_status.py*. Existing monitoring plugins may need adjustments.

Migration of old data is automated but may take some time depending on your environment.

7.7.2 Added

The App can now be configured in a way that it *does not* stop on the first error it encounters but instead continues to process the queue. This option is deactivated by default, meaning the behavior does not change. Note that we may eventually release another version and enable that feature.

7.8 v2.3.5

Released: 12. August 2025

7.8.1 Removed

The Resource manager field in the UMC has been removed. Ox resources still have that property in UDM, and can be edited or consulted.

There are no functional changes in the application.

7.9 v2.3.4

Released: 23. July 2025

7.9.1 Fixed

Fix OX Resources UDM handler preventing the OX Connector from working with UCS 5.2-2.

7.10 v2.3.3

Released: 10. June 2025

7.10.1 Fixed

UDM now actively prevents to create an OX Context with an ID already taken by another context.

7.10.2 Changed

Creating a new user in OX can now convert an existing OX guest account with the same e-mail address. Note that this requires OX 8.36.36, which is currently not present in the App Center. Also note that this only works for creating users; modifying users (or similar) will not have this feature. If the OX Connector cannot send this *convertguest* flag, the old behavior applies: Guest accounts with the same e-mail address as the user being processed will block further processing until the error is resolved.

7.11 v2.3.2

Released: 4. June 2025

7.11.1 Fixed

To keep track of certain object states, the App uses an internal key value store. The keys used are the distinguished names (DNs) of the LDAP objects. These have been stored as provided in the past. From now on, the DNs are normalized and lower-cased. Existing keys in that key value store are fixed during upgrade. For that, the new command *univention-app shell ox-connector rebuild-old.db* has been added.

When a user with the same name as a context admin was changed via UDM or UMC, there was an edge case which could lead to the OX connector changing an OX context admin in the OX database. This could lead to authorization issues for OX context admins and hence to synchronization issues for the OX connector.

7.11.2 Changed

In certain cases, the modification of an OX user was not immediately reflected on OX' side, so that the Connector may not get the correct database ID. The App now retries in these cases a few times, assuming that the data will eventually be retrievable.

7.12 v2.3.1

Released: 16. Apr 2025

7.12.1 Added

The administrator can now specify a log level for the ox-connector, see *App Settings* (page 15).

7.12.2 Changed

The app setting *OX_CONNECTOR_LOG_LEVEL* (page 18) is used to specify the log level of the ox-connector.

7.12.3 Fixed

The syntax for the *oxContext* attribute has been changed from string to integer.

7.13 v2.3.0

Released: 27. Mar 2025

7.13.1 Added

Allow provisioning of OX deputy permissions[23] through the **`OX Connector`** app. OX deputy permissions allow a user to act on behalf of another user in OX App Suite, providing delegated access to email and calendars. Administrators can configure these permissions to control the level of access and actions that deputies can perform.

The administrator can now add external ca-certificates to the container, see *Import additional CA certificates* (page 22).

7.13.2 Changed

The app setting *OX_IMAP_LOGIN* (page 16) can contain all attribute names as placeholders. While the connector still replaces the default value `{}` with the user's email address, you can set it to `{univentionObjectIdentifier}`, for example, given that such an attribute exists.

7.13.3 Fixed

Translations now work correctly.

7.14 v2.2.15

Released: 17. Feb 2025

7.14.1 Changed

Improved error message in case app settings *OX_USER_IDENTIFIER* or *OX_GROUP_IDENTIFIER* are not correctly configured.

Improve error message in case re-provisioning for a object is required.

[23] https://documentation.open-xchange.com/8/middleware/permissions_and_capabilities/deputy_permission.html

7.14.2 Fixed

The documentation wasn't explicit about setting the administrative password in the app settings for the *OX Connector App*.

The internal key value store for tracking user objects handles the key (dn) case insensitive now.

7.15 v2.2.14

Released: 29. Oct 2024

7.15.1 Changed

Allow special characters for the name of the access profile.

7.16 v2.2.13

Released: 17. Sep 2024

7.16.1 Changed

When changing an OX user in UDM, the OX attribute *default_sender_address* was left untouched. That is because it is a user preference, not "core data". Now, if an OX user is changed in UDM so that the *primaryMailAddress* changes and this has been the user's *default_sender_address*, it is overwritten to the new mail address - as this makes sense in next to all scenarios and would be considered an error if not done automatically.

7.17 v2.2.12

Released: 28. Aug 2024

7.17.1 Changed

You can now add LDAP containers to the list of default containers for functional accounts and select the container before creating a new functional account in UMC, see the *Functional accounts* (page 9) for more information.

7.18 v2.2.11

Released: 23. May 2024

7.18.1 Changed

Fixes a bug which prevents the removal of Open-Xchange contexts.

7.19 v2.2.10

Released: 26. April 2024

7.19.1 Changed

The performance of the OX Connector has been improved.

7.20 v2.2.9

Released: 12. April 2024

7.20.1 Added

It's now possible to change the attribute mapping between Open-Xchange and UCS through the script **`change_attribute_mapping.py`**. For more information, see *User attribute mapping* (page 20).

7.21 v2.2.8

Released: 16. January 2024

7.21.1 Changed

The *meta.db* also stores the error message and the filename that causes the error.

7.21.2 Added

The script *get_current_error.py* outputs a json with the contents of the *meta.db*. This json can be used to automate the app health checks.

The app settings *OX_USER_IDENTIFIER* and *OX_GROUP_IDENTIFIER* have been added. They give control over which UDM property is used as the unique identifier for users and groups in OX.

The script *check_sync_status.py* has been added. It can be used to identify data inconsistencies between UDM, OX and the listener files.

7.22 v2.2.7

Released: 7. September 2023

7.22.1 Changed

Allow any string in *OX_FUNCTIONAL_ACCOUNT_LOGIN_TEMPLATE* app setting to simplify SSO configurations.

7.22.2 Fixed

Fix *OX_FUNCTIONAL_ACCOUNT_LOGIN_TEMPLATE* empty app setting handling (Bug #56523).

Fix error in context change when modifying the context and the username in the same operation (Bug #56525).

7.23 v2.2.6

Released: 18. August 2023

7.23.1 Changed

The Functional Account login field is now configurable via the app setting *OX_FUNCTIONAL_ACCOUNT_LOGIN_TEMPLATE*.

7.24 v2.2.5

Released: 16. August 2023

7.24.1 Changed

User context change uses the *UserCopy* service.

7.25 v2.2.4

Released: 13. July 2023

7.25.1 Changed

The *imaplogin* field is now configurable via the app setting *OX_IMAP_LOGIN*.

7.26 v2.2.3

Released: 27. June 2023

7.26.1 Fixed

Corrected a typo in the *listener_trigger* script.

7.27 v2.2.2

Released: 22. June 2023

7.27.1 Fixed

The OX-Connector now prevents a scenario in which values set by users in the App Suite app were overwritten in a wrong way.

7.28 v2.2.1

Released: 07. June 2023

7.28.1 Changed

The OX-Context of a group is no longer modifiable in the groups module of UMC since the OX-Context of a group is always derived from the OX-Contexts of its users.

7.29 v2.2.0

Released: 01. June 2023

7.29.1 Changed

Removed use of old *oxDrive* and *oxAccessUSM* UDM properties. The OX Connector only uses the *oxmail/accessprofile* objects to control access rights.

The OX Connector does not require the *oxDisplayName* to be unique anymore.

The OX connector only sets a user's *default_sender_address*, *language*, and *timezone* when initially creating a user. Afterwards, any user can configure their settings in the OX App suite front-end.

The OX connector can handle user files in *listener/old/* without the *oxContext* attribute.

7.29.2 Deprecated

oxTimeZone and *oxLanguage* still exist as UDM attributes. But they are not evaluated anymore (see above in Changed; the Connector sets these attributes to the value set in the App Settings instead).

oxDisplayName still exists and is evaluated. At some later version, we will use the original *displayName* of a user.

7.30 v2.1.4

Released: 31. May 2023

This version has been revoked

7.31 v2.1.3

Released: 21. April 2023

7.31.1 Fixed

Changes to the *oxAccessUSM* attribute are now considered by the provisioning logic.

7.31.2 Changed

Added helper script to remove old listener files from users with empty *oxContextIDNum* attribute.

Removed *bindpwd* uses from *createextattr.py* script (#55985).

7.32 v2.1.2

Released: 4. April 2023

7.32.1 Changed

Changes in inst script for compatibility with App Center's OX App Suite.

7.33 v2.1.1

Released: 9. December 2022

7.33.1 Fixed

Fixed bug that prevented users from creating OX users from UMC.

7.34 v2.1.0

Released: 14. November 2022

7.34.1 Fixed

Remove the use of unnecessary *gid_ox* syntax for OX group names. All valid group names in UCS are now accepted in OX.

Avoid unnecessary group *change`* operation that can fail in large groups and lead to an infinite loop where the ox-connector tries to delete an already deleted user.

Change *oxcontext contextid* syntax from string to integer.

7.34.2 Changed

Refactor of internal project structure.

Update of scripts and internal files.

7.34.3 Added

Prepare support for Univention OX App suite.

7.35 v2.0.1

Released: 9. September 2022

7.35.1 Fixed

Avoid unnecessary look-ups in the OX database when syncing groups: Users that appear to not be present in the database will be treated as such instead of double checking.

Avoid 500 log messages in OX by guarding user look-ups by an *exists* call.

7.36 v2.0.0

Released: 26. April 2022

7.36.1 Added

With OX App Suite 7.10.6 Open-Xchange added *Functional Mailboxes* to OX App Suite, see *OX App Suite - Minor Release v7.10.6 - Feature Overview* [7]. OX App Suite shares functional mailboxes among other users in the same context.

For more information, see *Functional accounts* (page 9).

7.37 v1.1.0

7.37.1 Added

OX App Suite knows access and can grant them individually to users. The **`OX Connector`** app supports *access profiles* through the file `ModuleAccessDefinitions.propertiers`.

The connector generates the file locally on the UCS system each time an administrator modifies objects in the UDM module `oxmail/accessprofile`. It doesn't provision the data to OX App Suite directly. The connector uses the *access profiles* and sets the attribute `oxAccess` during provisioning.

For limitations, see *No plausibility validation in access profile rights* (page 28).

BIBLIOGRAPHY

[1] *App Suite Admin Guide 7.10.5*. OX Software GmbH, 2022. URL: https://oxpedia.org/wiki/index.php?title=AppSuite:AdminGuide_7.10.5.

[2] *UCS 5.2 Manual*. Univention GmbH, 2025. URL: https://docs.software-univention.de/manual/5.2/en/.

[3] Raphaël Hertzog and Roland Mas. *The Debian Administrator's Handbook*, chapter Shell and Basic Commands. Freexian SARL, First edition, 2020. URL: https://www.debian.org/doc/manuals/debian-handbook/short-remedial-course.en.html#sect.shell-and-basic-commands.

[4] *Referential Integrity in OpenLDAP Software 2.4 Administrator's Guide*. The OpenLDAP Project, March 2021. URL: https://openldap.org/doc/admin24/overlays.html#Referential%20Integrity.

[5] *App Suite Context management*. OX Software GmbH, 2020. URL: https://oxpedia.org/wiki/index.php?title=AppSuite:Context_management.

[6] *App Suite Resource management*. OX Software GmbH, 2015. URL: https://oxpedia.org/wiki/index.php?title=AppSuite:Resource_management.

[7] *OX App Suite - Minor Release v7.10.6 - Feature Overview*. OX Software GmbH, December 2021. URL: https://www.open-xchange.com/hubfs/Imported%20files/Feature_Overview_OXAppSuite_7_10_6.pdf?hsLang=en.

INDEX

M

O

P

R

S

T

U