univention
be open.

OX Connector app

Release 2.2.12

Univention GmbH

Aug 27, 2024

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 11) with a reference list about the app settings of the OX Connector app.
3. *Architecture* (page 17) of the app, how the connector works and the connector cache.
4. *Limitations* (page 21) of the app.
5. *Troubleshooting* (page 23) about log files, health check, queuing and rebuild the cache.
6. *Changelog* (page 33) 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 5.0 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 5.0 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/5.0/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 18).

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 21).

   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 12) and *OX_MASTER_PASSWORD* (page 12).

   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
   ```

   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.

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

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 5.0 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 11).
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`.

For more information, see Delegated administration for UMC modules[6] in *UCS 5.0 Manual* [2].

[4] https://docs.software-univention.de/manual/5.0/en/domain-ldap/system-roles.html#system-roles
[5] https://docs.software-univention.de/manual/5.0/en/software/app-center.html#software-appcenter
[6] https://docs.software-univention.de/manual/5.0/en/central-management-umc/delegated-administration.html#delegated-administration

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 11). 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`**.

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 5.0 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.

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*.

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)*.

[7] https://docs.software-univention.de/manual/5.0/en/central-management-umc/delegated-administration.html#delegated-administration

5. Click *Save*.

See also:

User management[8] in *UCS 5.0 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.

To create a group:

2. Click *Add* to create a group.
3. On the *General* tab, fill out the required fields and add users as group members.
4. Go to the *OX App Suite* tab and activate the *Activate Group in OX*.
5. Click *Create group*.

To update a group:

2. Click a group to edit.
3. 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.
4. Click *Save*.

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

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

To remove a group from OX App Suite:

2. Click a group to edit.
3. Go to the *OX App Suite* tab and clear the checkbox *Activate Group in OX*.
4. Click *Save*.

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

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

See also:

Group management[9] in *UCS 5.0 Manual* [2].

[8] https://docs.software-univention.de/manual/5.0/en/user-management/index.html#users-general
[9] https://docs.software-univention.de/manual/5.0/en/groups.html#groups

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 22).

2.5 Functional accounts

New 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.

2.5.1 Default LDAP position for functional accounts

New 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.

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 11) 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 Docker container needs the root certificate for validation.

For example, to add a custom certificate, run the following commands on the UCS system, where OX Connector is installed:

```
$ univention-app shell ox-connector
/oxp # wget --no-check-certificate \
  https://ox-app-suite.example.com/root-ca.crt \
  -O /usr/local/share/ca-certificates/ox-app-suite.crt
/oxp # update-ca-certificates
"WARNING: ca-certificates.crt does not contain exactly one certificate or CRL:␣
↪skipping"
```

Administrators can ignore the warning.

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

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_IMAP_LOGIN` can be set to `'{}*dovecotadmin'`. The curly braces are used as a template for the primary mail address. The resulting *imaplogin* value would then look like this:

```
myuser@maildomain.de*dovecotadmin
```

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** 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** 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.

3.2 Univention Configuration Registry variables

ox/context/id

The app setting *DEFAULT_CONTEXT* (page 12) sets the value of the Univention Configuration Registry variable *ox/context/id* (page 14).

Upon installation of the app **OX Connector**, the OX Connector creates the extended attribute `oxContext` and uses the value from *ox/context/id* (page 14) 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 14) **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 12) does **neither** change *ox/context/id* (page 14) **nor** the extended attribute `oxContext`.

3.3 User attribute mapping

New 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 3.1 shows how to set the mapping of the Open Xchange property `userfield01` to the UDM property `description`.

Listing 3.1: 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 15) arguments multiple times in the same invocation. Listing 3.2 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 3.2: 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 3.3: 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 3.4 shows an example to set the theoretical attributes `CustomAttributeUserMail` and `CustomAttributeUserMail2` as alternatives to the Open-Xchange property `email1`.

Listing 3.4: 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 3.5: 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}")

  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
```

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.

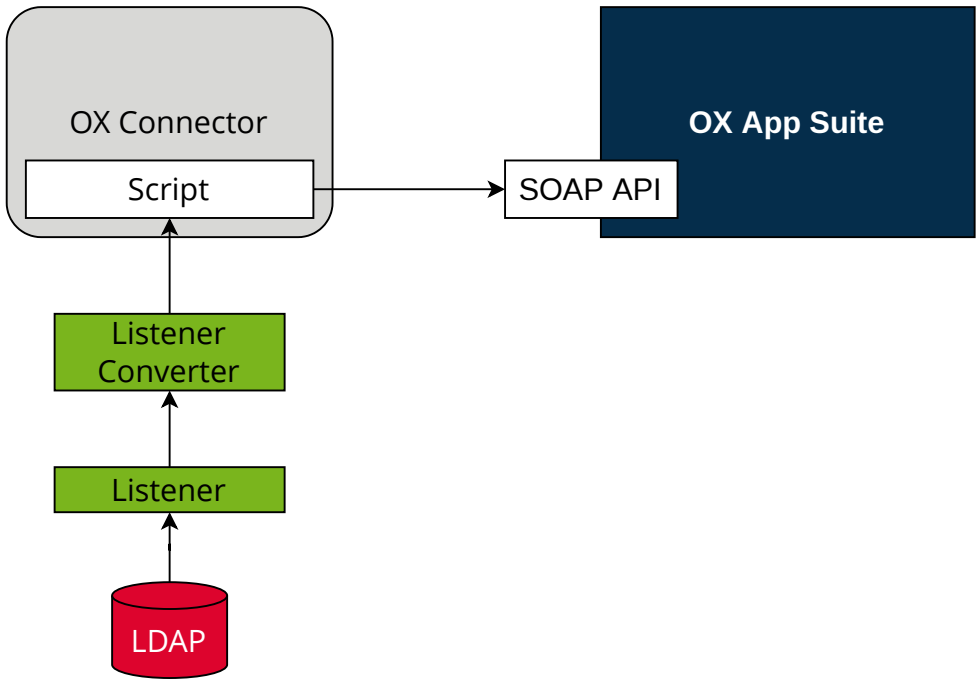

Fig. 4.1: OX Connector app architecture
View focuses on the elements LDAP Directory, Listener, Listener Converter, OX Connector with the provisioning script, OX App Suite, and its SOAP API.

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[10] in *UCS 5.0 Manual* [2].

[10] https://docs.software-univention.de/manual/5.0/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 `entryUUID` 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 `entryUUID`.

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*, processes it and sends data to the *SOAP API*.

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

It exits upon the first failed *SOAP API* request and repeats once the *Listener Converter* triggers the *Script*.

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

SOAP API
OX App Suite uses SOAP[11] 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.

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

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*.

4.2.2 Provisioning

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

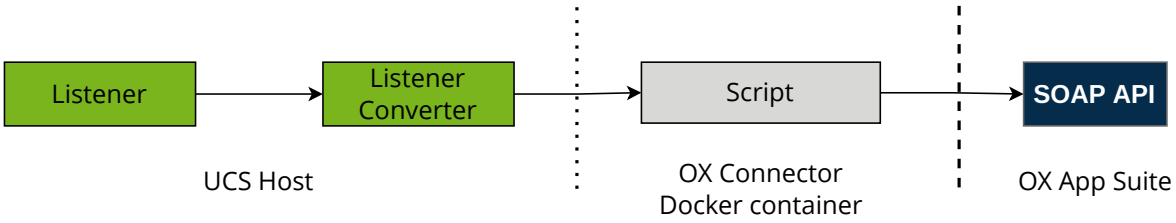

Fig. 4.2: Provisioning procedure

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* iterates over 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 the at step 2.

For more information about the file contents of the *Listener* and *Listener Converter*, see *Overview* (page 17).

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 Cache

New in version 2.0.0.

OX App Suite creates an *internal ID* for every user object it creates or updates. The OX Connector saves this *internal ID* in the JSON files, when it processed the objects without errors. The connector doesn't store that ID in the UCS LDAP directory, but maintains a file based cache on *internal ID*s created by OX App Suite.

The directory for the JSON files is `var/lib/univention-appcenter/apps/ox-connector/data/listener/old/`.

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 cache.

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 does support the use of **OX Connector** towards the **OX App Suite** app from Univention App Center. The **OX Connector** takes over the provisioning, the **OX App Suite** ships the actual groupware.

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

If not, you find the password on the UCS system that runs **OX App Suite** in the file `/etc/ox-secrets/master.secret`. The username of the administrative account is `oxadminmaster`. You can set the credentials in the app settings of the **OX Connecor**, see *Configuration* (page 11).

5.2 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 logs the filename with its path in the *Listener Converter* log file, see *Log files* (page 23).

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 25). After the conflict resolution, the connector continues to process the provisioning queue.

Design decision

The OX Connector doesn't provide logic to resolve conflicts automatically, because the conflict causes can vary a lot. For example, when connector would ignore the conflict and continue, a later operation may refer to the ignored item. The connector can't complete it, because the current queue item refers to a previous, unprocessed item. The OX Connector could ignore the next conflict again, and again. The ignores pile up unresolved conflicts that can lead to a heavy conflict or a serious problem with the user provisioning without any relation to the actual root cause. Administrators would have quite a hard job to resolve the conflict.

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[12].

[12] 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*.

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 Health check

First, have a look at the log file for the *Listener Converter* and look for warnings and errors, see *Log files* (page 23).

The OX Connector has a good health when the number of files for provisioning for the *Listener* and the *Listener Converter* is low. For a quick verification, run the following command on the UCS system with the OX Connector installed:

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
```

Listing 6.2: Verify the number of unprocessed files for the *Listener Converter*.

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

The *Listener Converter* logs consecutive errors in the log file, for example:

```
INFO    This is consecutive error #{some number}
```

Such entries indicate that the provisioning has issues with processing the queue. For more information, see *Queuing* (page 25).

You can use the script *get_current_error.py* to automate the health check on your preferred monitoring system.

```
univention-app shell ox-connector /usr/local/share/ox-connector/resources/get_
↪current_error.py
```

This script outputs a json with some information about the current state of the OX Connector.

If there is an error:

```
{'errors': '10', 'message': "HTTPSConnectionPool(host='ucs11.ucs.net', port=443):␣
↪Max retries exceeded with url: /webservices/OXContextService?wsdl (Caused by␣
↪NewConnectionError('<urllib3.connection.HTTPSConnection object at 0x7f7b1083a610>
↪: Failed to establish a new connection: [Errno 111] Connection refused'))",
↪'filename': '/var/lib/univention-appcenter/apps/ox-connector/data/listener/2023-
↪12-11-11-22-22-856263.json'}
```

If the ox-connector is working:

```
{'errors': '0'}
```

The script *get_current_error.py* can easily be integrated into a Nagios plugin script, as shown in the following example:

```
#!/bin/bash

nagiosCheck () {
    result=$(/var/lib/univention-appcenter/apps/ox-connector/data/resources/get_
↪current_error.py)
    status=$(echo ${result} | jq ' if .errors == "0" then 0 else 1 end')

    case $status in
    0)
        echo "OK: No errors found."
        exit 0
        ;;
    1)
        error_msg=$(echo ${result} | jq ' .message ')
        error_file=$(echo ${result} | jq ' .filename ')
        echo "WARNING: ${error_msg}. This error is caused by the listener file $
↪{error_file}"
        exit 1
        ;;
    esac
}

nagiosCheck
```

6.3 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 23) 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 delete the flawed file. The log file reveals the flawed file and its path, see *Delete one item from the queue* (page 25).

6.4 Queuing

The queue for provisioning consists of JSON files. *How the connector works* (page 18) describes the connector's data processing. Administrators can manually intervene with the queue in the following cases.

6.4.1 Delete one item from the queue

Administrators can remove an item in the queue, if the connector can't process it and interrupts the provisioning process. The connector retries to provision this item and continually fails.

To find and remove the problematic item from the queue, follow these steps:

1. Open the log file of the *Listener Converter*. For the log file location, see *Log files* (page 23).
2. Find the filename of the item that the *Listener Converter* retries to provision. For example, the log file shows:

   ```
   Error while processing /var/lib/univention-appcenter/apps/ox-connector/data/
   ↪listener/$timestamp.json
   ```

   `$timestamp` has the format `%Y-%m-%d-%H-%M-%S`.
3. Remove the problematic item:

   ```
   $ rm /var/lib/univention-appcenter/apps/ox-connector/data/listener/$timestamp.
   ↪json
   ```

6.4.2 Re-provision one specific UDM object

The OX Connector allows to re-provision one UDM object to OX App Suite. The following snippet provisions one user object:

Listing 6.3: Re-provision one UDM object

```
DN="uid=user100,cn=users,$(ucr get ldap/base)"
ENTRY_UUID="$(univention-ldapsearch -b "$DN" + | grep entryUUID | awk '{ print $2 }
↪')"
cat > /var/lib/univention-appcenter/listener/ox-connector/$(date +%Y-%m-%d-%H-%M-
↪%S).json <<- EOF
{
    "entry_uuid": "$ENTRY_UUID",
    "dn": "$DN",
    "object_type": "users/user",
    "command": "modify"
}
EOF
```

6.4.3 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 6.4: 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 Rebuild cache

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 *Cache* (page 20).

To rebuild the cache, run the following commands:

Listing 6.5: Rebuild cache for *internal ID*

```
$ univention-app shell ox-connector
/oxp # update-ox-db-cache --delete
/oxp # update-ox-db-cache
```

Changed in version 2.0.0: Rebuild the cache after an update to version 2.0.0, because previous versions didn't maintain the cache for the *internal ID*.

Otherwise, the OX Connector app falls back into the much slower mechanism and runs a database query per user during the provisioning.

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.

`update-ox-db-cache --build-cache` can speed up the rebuild, because it retrieves all users of a 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)
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:

```
2023-11-17 09:21:20 INFO    Loading old object from /var/lib/univention-appcenter/
↪apps/ox-connector/data/listener/old/d52a12f0-2d89-103c-82b6-b945bc689f52.json
2023-11-17 09:21:20 INFO    Loading old object from /var/lib/univention-appcenter/
↪apps/ox-connector/data/listener/old/f029fd00-8247-103c-89e3-bd95c6adf546.json
2023-11-17 09:21:20 INFO    Error while processing /var/lib/univention-appcenter/
↪apps/ox-connector/data/listener/2023-02-27-13-30-03-471251.json
2023-11-17 09:21:20 WARNING Traceback (most recent call last):
2023-11-17 09:21:20 WARNING   File "/tmp/univention-ox-connector.listener_trigger",
↪ line 341, in run_on_files
2023-11-17 09:21:20 WARNING     function(obj)
2023-11-17 09:21:20 WARNING   File "/usr/lib/python3.9/site-packages/univention/ox/
↪provisioning/__init__.py", line 103, in run
2023-11-17 09:21:20 WARNING     for new_obj in get_group_objs(obj):
2023-11-17 09:21:20 WARNING   File "/usr/lib/python3.9/site-packages/univention/ox/
↪provisioning/__init__.py", line 156, in get_group_objs
2023-11-17 09:21:20 WARNING     user_obj = univention.ox.provisioning.helpers.get_
↪old_obj(user)
2023-11-17 09:21:20 WARNING   File "/tmp/univention-ox-connector.listener_trigger",
↪ line 72, in _get_old_object
2023-11-17 09:21:20 WARNING     return object_from_path(path_to_old_user)
2023-11-17 09:21:20 WARNING   File "/tmp/univention-ox-connector.listener_trigger",
↪ line 261, in object_from_path
2023-11-17 09:21:20 WARNING     entry_uuid = content["id"]
2023-11-17 09:21:20 WARNING TypeError: 'NoneType' object is not subscriptable
2023-11-17 09:21:20 INFO    This is consecutive error #18
2023-11-17 09:21:20 INFO    Sleeping for 0 sec
2023-11-17 09:21:20 INFO    Successfully processed 0 files during this run
```

You can check which users are missing in the old directory by running the next command. It will print the *DN* of the users that need to be provisioned again. Then you can follow the instructions here *Re-provision one specific UDM object* (page 25) to synchronize the missing users.

```
univention-ldapsearch "(&(univentionObjectType=users/user)(isOxUser=OK))"␣
↪entryUUID | sed -ne 's/entryUUID: //p' | xargs -I{} bash -c "test -e  /var/lib/
↪univention-appcenter/apps/ox-connector/data/listener/old/{}.json || univention-
↪ldapsearch -LLL  entryUUID={} 1.1"
```

6.8 Verify data consistency

In OX Connector version 2.2.8 a new script called *check_sync_status.py* can be used to verify that the data in *UDM*, the listener/old directory and the OX database are the same. If the App settings *OX_USER_IDENTIFIER* (page 13), *OX_GROUP_IDENTIFIER* (page 14), *OX_FUNCTIONAL_ACCOUNT_LOGIN_TEMPLATE* (page 13), *OX_IMAP_LOGIN* (page 12) are set to non default values, the script can detect and report inconsistencies between the OX database, listener files and UDM.

```
$ univention-app shell ox-connector
/oxp # ./check_sync_status.py --dn uid=qwert,cn=users,dc=test,dc=ucs --udm_admin_
↪account administrator --udm_password_file udm.secret --udm_host https://master.
↪master.ucs
```

Note:

./check_sync_status.py –help

--dn DN Check the object with the specified dn

--udm_module UDM_MODULE Object's udm module. Required if the property is missing in the old/ directory object.

--ox_context OX_CONTEXT Object's ox context. Required if the property is missing in the old/ directory object.

--resync Re-sync object data by creating a new file in the listener. Re-synchronizing groups will only work if its users are correctly provisioned.

--udm_admin_account UDM_ADMIN_ACCOUNT Udm user used for connection.

--udm_password_file UDM_PASSWORD_FILE Udm password

--udm_host UDM_HOST Udm host

6.9 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[13].
- Provide the relevant messages and tracebacks from *Log files* (page 23), 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.

6.10 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/
```

(continues on next page)

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

(continued from previous page)

```
↪soap/backend.py", line 176, in modify
2024-01-11 13:57:39 WARNING     assert self.name is not None
```

```
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/
↪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)
```

(continues on next page)

(continued from previous page)

```
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
```

CHAPTER

SEVEN

CHANGELOG

This changelog documents all notable changes to the OX Connector app. Keep a Changelog[14] is the format and this project adheres to Semantic Versioning[15].

7.1 2.2.12

Released: 28. Aug 2024

7.1.1 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.

7.2 2.2.11

Released: 23. May 2024

7.2.1 Changed

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

7.3 2.2.10

Released: 26. April 2024

7.3.1 Changed

The performance of the OX Connector has been improved.

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

7.4 2.2.9

Released: 12. April 2024

7.4.1 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 14).

7.5 2.2.8

Released: 16. January 2024

7.5.1 Changed

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

7.5.2 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.

7.6 2.2.7

Released: 7. September 2023

7.6.1 Changed

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

7.6.2 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).

7.7 2.2.6

Released: 18. August 2023

7.7.1 Changed

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

7.8 2.2.5

Released: 16. August 2023

7.8.1 Changed

User context change uses the *UserCopy* service.

7.9 2.2.4

Released: 13. July 2023

7.9.1 Changed

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

7.10 2.2.3

Released: 27. June 2023

7.10.1 Fixed

Corrected a typo in the *listener_trigger* script.

7.11 2.2.2

Released: 22. June 2023

7.11.1 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.

7.12 2.2.1

Released: 07. June 2023

7.12.1 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.

7.13 2.2.0

Released: 01. June 2023

7.13.1 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.

7.13.2 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.

7.14 2.1.4

Released: 31. May 2023

This version has been revoked

7.15 2.1.3

Released: 21. April 2023

7.15.1 Fixed

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

7.15.2 Changed

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

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

7.16 2.1.2

Released: 4. April 2023

7.16.1 Changed

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

7.17 2.1.1

Released: 9. December 2022

7.17.1 Fixed

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

7.18 2.1.0

Released: 14. November 2022

7.18.1 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.

7.18.2 Changed

Refactor of internal project structure.

Update of scripts and internal files.

7.18.3 Added

Prepare support for Univention OX App suite.

7.19 2.0.1

Released: 9. September 2022

7.19.1 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.

7.20 2.0.0

Released: 26. April 2022

7.20.1 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).

7.21 1.1.0

7.21.1 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 22).

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 5.0 Manual*. Univention GmbH, 2021. URL: https://docs.software-univention.de/manual/5.0/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

Q

R

S

U