

# Univention Active Directory Connector

Topic:	Installation and configuration of Univention Active Directory Connector
Date:	4. März 2009
Pages:	<a href="#">15</a>
Version number:	2895
Autoren:	Univention GmbH   <a href="mailto:feedback@univention.de">feedback@univention.de</a>

## Inhaltsverzeichnis

<b>1</b>	<b>Introduction to the Univention Active Directory Connector</b>	<b>3</b>
<b>2</b>	<b>Installation</b>	<b>3</b>
2.1	Active Directory . . . . .	3
2.2	Active Directory users . . . . .	5
2.3	Univention Corporate Server . . . . .	6
<b>3</b>	<b>Mode of operation</b>	<b>7</b>
3.1	Synchronisation process . . . . .	7
<b>4</b>	<b>Mapping configuration</b>	<b>8</b>
<b>5</b>	<b>Preconfigured mappings</b>	<b>10</b>
5.1	Container and organisational units . . . . .	10
5.2	Groups . . . . .	11
5.3	Users . . . . .	12
<b>6</b>	<b>Univention Configuration Registry variables</b>	<b>13</b>
6.1	Basic configuration . . . . .	13
6.2	Predefined mapping definitions . . . . .	14
6.3	Synchronisation with Windows 2000 . . . . .	14
<b>7</b>	<b>Tools</b>	<b>15</b>
7.1	univention-adsearch . . . . .	15
7.2	univention-connector-list-rejected . . . . .	15

# 1 Introduction to the Univention Active Directory Connector

The Univention Active Directory Connector (AD Connector in short) makes it possible to synchronise directory service objects between a Windows 2000/2003 server under Active Directory (AD) and Univention Corporate Server.

The synchronisation settings can be defined individually, making it possible for the administrator to control synchronisation very accurately by just synchronising certain objects and attributes.

In the default setting, containers, organisational units, users, and groups are synchronised. Users have an exceptional position since the password cannot be queried via the LDAP protocol in Active Directory. A special service is installed on the Windows server for this purpose, which enables password synchronisation. The client accounts are not synchronised in the default configuration since Windows clients can only be included in one domain, so they cannot be simply adopted from an Active Directory environment into a Windows NT domain mapped via UCS.

Users can make access to services of both environments in a transparent way; this is due to the possibility of having the same user settings in both domains. After logging into a UCS domain, a subsequent connection to a file share or to an Exchange server with Active Directory is possible without a renewed password request.

Users and administrators will find users and groups of the same name on the resources of the other domain, and can thus work with their familiar permission structures.

The Univention AD Connector can only be installed on a DC master or DC backup system since only in such environments are the complete data present in the LDAP. Despite intensive tests, the possibility of the results of the synchronisation procedure affecting the operation of a productive domain cannot be excluded. Therefore tests with regard to the Connector's requirements in individual cases should be performed in a separate environment.

## 2 Installation

### 2.1 Active Directory

In Active Directory a certificate for the server has to be created and exported so that encrypted communication becomes possible. Thereafter the password service is to be installed and started.

#### 2.1.1 Certificate

If the certificate service is not already installed then this can be subsequently done via: Start -> Settings -> Control Panel -> Software -> Windows Components,

Select certificate service -> Continue, Select the trusted root certification authority for the company -> Continue, Enter domain name -> Continue -> Continue.

This certificate has to be exported and copied to the UCS system: Certification authority -> AD domain -> Properties -> Show certificate -> Details -> Copy to file -> DER encoded binary X.509.

On the UCS system, the certificate should then be converted via openssl since this is the required format:

```
openssl x509 -inform der -outform pem -in <infile.cer> -out <outfile.pem>
```

The file name of the certificate is subsequently stored in the Univention Configuration Registry registry in PEM format:

```
univention-config-registry set \  
connector/ad/ldap/certificate=/etc/univention/ad.pem
```

### 2.1.2 Password service

Active Directory prohibits password queries via the LDAP protocol thus necessitating the installation of a package on the Windows server. The package is called `ucs-ad-connector.msi` and can be found on the UCS system at </usr/share/univention-ad-connector/> as soon as the **univention-ad-connector** package is installed. Installation of the package is started by double-click.

The package is installed automatically into the directory `C:\Windows\UCS-AD-Connector`

In addition, the password service is integrated as a system service into the Windows environment thereby making it possible for this service to be started either automatically or manually.

After a certificate has been created for the client on the UCS system (see chapter 2.3.2), the files **private.key** and **cert.pem** should be copied from the UCS certificate directory (</etc/univention/ssl/FQDN>) to the installation directory of the password service.

The password service can then be started via **Start -> Administration -> Services**. If the service is to be installed in a different directory, it can subsequently be copied to this new location. First the service should be stopped and then removed. To do so the following commands can be entered at the DOS prompt:

```
C:\Windows\AD-Connector\ucs-ad-connector.exe -stop  
C:\Windows\AD-Connector\ucs-ad-connector.exe -remove
```

Then the UCS AD Connector directory can be moved and the service can be initialised anew.

```
C:\AD-Connector\ucs-ad-connector.exe -install  
C:\AD-Connector\ucs-ad-connector.exe -start
```

## 2.2 Active Directory users

Apart from the user **Administrator**, who is initially provided with sufficient privileges in Active Directory, other user accounts can also be used for the password service or for LDAP access. For this purpose, sufficient privileges are to be assigned to the respective users.

Configuration procedures will therefore be exemplified by the user account **Administrator** in later chapters of this manual, for this account is usually provided with sufficient privileges for running the AD Connector. If however the privileges are too extensive, i.e. if there is merely a unidirectional synchronisation to be performed, different accounts in the form of replication users can be created for accessing the LDAP or operating the password service.

### 2.2.1 User account for LDAP Replication

Every authenticated user within the standard configuration (Windows 2003) has sufficient read access to the LDAP of an Active Directory for usage in the AD Connector. Containers and organisational units as well as all users and groups can be read. Write access, which is necessary for aligning UCS accounts to the AD, is only assigned to members of the **Administrators** group.

In the extended view of the MMC plugin **Active Directory users and clients**, the permissions for replication users can be assigned via the properties of organisational units. If an account which was created for replication has its permissions for an organisational unit withdrawn, the subordinate users and groups will no longer be synchronised according with the UCS.

If such restricted read access is defined, "rejects" might occur in cases where not all the required privileges for users and groups are fulfilled for synchronisation. A typical reason could be a primary group in a container, which can no longer be read by the connector. If, for example, read access to the standard group container is disabled, AD users with the primary group **Domain Users** located in this container can no longer be created in the UCS directory. In such a case, a different primary group should be assigned in AD or the group should be moved.

Appropriate restrictions should be defined in the UCS mapping in order to avoid unnecessary rejects; then the connector will not try to synchronise data to AD where it has no write access. In case of a read-only access to AD, the `sync_mode` for all objects within the UCS mapping definition can be set to read (see chapter 4).

### 2.2.2 User account for password service

The password service to which the AD Connector makes access, clearly needs more privileges for read access to the SAM database than a standard user. If a different user with substantially restricted rights is utilised for LDAP access to AD, then this user cannot be utilised at the same time for operating the password service.

After its installation, the password service is started as a local system service and initially is not assigned to any user account. If a specific user is to be allocated to this service, the user account created for this purpose can be added to the group of Administrators in order to preserve all the necessary privileges. In addition, this user has to be authorised to have read access to the installation directory `C:\Windows\UCS-AD-Connector`, and write access to the files `copypwd.txt` and `copypwd.in.txt` as well as the log file within this directory.

A different user may start the service manually if he's granted permissions in the local security policy for local login and debugging programs. Unfortunately these permissions are not utilised to a sufficient extent if the program is automatically started as a service.

## 2.3 Univention Corporate Server

### 2.3.1 Installation

Installation is either carried out when installing the UCS or subsequently via the command:

```
apt-get install univention-ad-connector
```

The basic configuration is carried out via Univention Configuration Registry. It is compulsory to set the following variables (a complete list can be found in chapter 6):

- `connector/ad/ldap/host`  
This variable contains the FQDN of the Active Directory Servers, e.g. ***w2k3.ad.univention.de***.
- `connector/ad/ldap/base`  
The LDAP base DN of the Active Directory Servers, e.g. ***dc=ad,dc=univention,dc=de***.
- `connector/ad/ldap/binddn`  
This LDAP user is utilised by the Univention AD Connector for carrying out changes in the LDAP of the Active Directory, e.g. ***cn=Administrator,cn=users,dc=ad,dc=univention,dc=de***.
- `connector/ad/ldap/bindpw`  
The file which contains the password of the user referenced in the Univention Configuration Registry-Variable `connector/ad/ldap/binddn`, e.g. [\*/etc/univention/ad.secret\*](#).
- `connector/ad/ldap/certificate`  
The certificate of the Windows server in PEM format, see chapter 2.1.1, e.g. [\*/etc/univention/ad.pem\*](#).
- `connector/ad/windows_version`  
The formats of LDAP database queries differ between Windows 2000 and Windows 2003. To be able to synchronise against an AD from Windows 2000, this Univention Configuration Registry variable has to be configured to ***win2000***.

### 2.3.2 Certificate

A certificate for encrypted communication with the password service on Active Directory has to be created on the UCS system. For this purpose a member server can be created with the Univention Directory Manager web front-end, thus automatically creating a certificate, or a certificate can be created manually via the command line tool `univention-certificate`. The generated certificate will be located in the directory `/etc/univention/ssl/<FQDN>`. The two files `private.key` and `cert.pem` are to be copied to the installation directory of the password service on the Windows server (see chapter 2.1.2).

### 2.3.3 Starting/stopping the service

The Univention AD Connector service can be started by using the command `/etc/init.d/univention-ad-connector start`, and stopped by the command `/etc/init.d/univention-ad-connector stop`.

## 3 Mode of operation

### 3.1 Synchronisation process

After the initial start of the connector, an initialisation run is carried out. During this process all entries in the UCS are read and converted in accordance with the specified mapping into AD objects; these objects are then added on the UCS side or modified, should they already exist. Afterwards all objects in AD are read, converted into UCS objects, and added or modified accordingly. As long as changes are still available, queries to the directory service servers are continued.

When no further changes are available, a waiting period occurs until the Univention AD Connector starts querying the Active Directory or the local OpenLDAP again. This period can be specified via the Univention Configuration Registry-Variable `connector/ad/poll/sleep`. The unit used is seconds; after installation this value is set to five seconds.

If an object cannot be synchronised, then it is rejected. When starting the Univention AD Connector, an attempt is made to replay these objects once more. The same attempt is made again after several runs. This cycle can be controlled via the Univention Configuration Registry-Variable `connector/ad/retryrejected`. By default, ten cycles have to pass before the objects can be replayed.

Logging and status messages are stored in the files

```
/var/log/univention/connector-status.log
/var/log/univention/connector.log
/var/log/univention/connector-tracebacks.log
```

The debug level can be changed via the Univention Configuration Registry-Variable `connector/debug/level`.

The DNs of the unsynchronised objects can be queried via the script `univention-connector-list-rejected`. It is recommended to stop the connector for this purpose. General information on these objects (depending on the debug level) can be found in the file `connector.log`; more specified error messages generated by LDAP access processes to Active Directory or the Univention Directory Manager can be found in the file `connector-tracebacks.log`.

## 4 Mapping configuration

The file `/etc/univention/connector/ad/mapping.py` contains definitions of the objects and attributes to be synchronised. This file is written in the Python scripting language, which allows highly flexible definitions. Management of this file is handled via Univention Configuration Registry; changes should, therefore, be carried out at the corresponding template files (`/etc/univention/connector/ad/mapping`). Since the template interprets Univention Configuration Registry variables for certain standard options, direct intervention is not always necessary. The existing Univention Configuration Registry variables are explained in chapter 6.2.

The variable `global_ignore_subtree` is used for configuring a list of containers to be ignored. Univention Configuration Registry variables can be used in this context according to the template mechanism, e.g.:

```
global_ignore_subtree=[ 'cn=univention,@\%@ldap/base@%\%',  
                        'cn=System,@\%@connector/ad/ldap/base@%\%' ]
```

These instructions cause the container ***cn=univention*** on the UCS side and the container ***cn=System*** with all its sub-objects on the Active Directory side to be ignored.

All further mapping options are defined in the ***ad\_mapping*** dictionary. The keys in this context are always unique names, e.g. ***user***. A ***univention.connector.property*** object is passed to this key as a value. This object has the following properties:

***ucs\_module*** The Univention Directory Manager module which is utilised for processing this object. A list of all the possible modules can be accessed via the command `univention-directory-manager help`.

***sync\_mode*** The synchronisation mode to be used. Possible values are ***read***, ***write***, ***sync*** and ***none***. ***Read*** means the objects are replicated from Active Directory to the UCS; ***write*** means the objects are replicated from the UCS to Active Directory; ***sync*** means the data are synchronised; ***none*** means no changes are carried out. In the predefined mapping, the global sync mode of the Univention Configuration Registry-Variable `connector/ad/mapping/syncmode` is used.

**scope** States the search depth for the LDAP search processes. Possible values are **sub**, **one** and **base**.

**con\_search\_filter** The LDAP search filter for identifying the objects in Active Directory.

**match\_filter** The LDAP search filter for identifying the objects in the UCS.

**ignore\_filter** A filter for objects to be ignored can be specified here.

**ignore\_subtree** The containers in this list are ignored, as are the corresponding sub-objects.

**con\_create\_objectclass** A list of object classes utilised on the Active Directory side.

**dn\_mapping\_function** A list of functions called for converting the DN from Active Directory to UCS and vice versa. This setting for example is required for users since on the UCS side, the **uid** attribute is used and on the Active Directory side the **cn** attribute is used in the DN.

**attributes** A dictionary of **univention.connector.attribute** objects which are directly processed during creation and whilst changes are taking place.

**ucs\_create\_functions** A list of functions processed after the creation of an object.

**post\_con\_modify\_function** A list of functions processed after the modification of an object in Active Directory.

**post\_ucs\_modify\_functions** A list of functions processed after the modification of an object in UCS.

**post\_attributes** A dictionary of attributes, which can only be changed in a second step rather than directly during the creation of an object.

**mapping\_table** A dictionary whose keys correspond to the attribute keys. The key is assigned a list of string-tuples containing UCS and AD designators respectively. If during synchronisation a UCS object from this list is found, the corresponding AD value is set and vice versa. This property is used, for example, for the mapping of group names.

**position\_mapping** Assigns a sub-container in the UCS to a sub container in AD. This property should be used with care in case there are containers of identical names on the other side in each case. For example, the groups of the UCS containers **cn=groups** should not be mapped to the standard AD container **cn=users**, since this container already exists in the UCS. Groups created in the UCS under **cn=users** would consequently not lie within this relation and could be the cause of errors.

## 5 Preconfigured mappings

Generally, synchronisation is effected to the exclusion via filters of ignored containers. These are the following sub-containers of the LDAP base:

On the UCS side:

```
cn=univention
cn=policies
cn=shares
cn=printers
cn=networks
cn=kerberos
cn=dhcp
cn=dns
cn=computers
```

On the AD side:

```
cn=System
cn=Builtin
cn=ForeignSecurityPrincipals
ou=Domain Controllers
cn=Program Data
```

### 5.1 Container and organisational units

The structuring elements are synchronised together with their descriptions. In addition, the containers **cn=mail** and **cn=kerberos** are ignored on both sides. Several particularities on the AD side are to be considered with containers. Active Directory does not offer any means of creating containers in the **AD user and group management**, yet the containers are displayed in the extended mode (**View → Extended functions**).

### Particularities

- Active Directory cannot create organisational units below containers; this is why OUs created in this way in the UCS as well as the corresponding subordinated objects are not synchronised.
- Containers or organisational units deleted in AD will be erased recursively in UCS, meaning that unsynchronised sub-objects not visible in AD might also be removed.

## 5.2 Groups

Groups are synchronised by means of their group names; the primary group of the user is considered in this process (in AD, this group is only lodged at the user in LDAP). Group members having no counterpart in the other system, due to ignore filters for example, will be ignored (meaning they remain members of the group). In addition, the description of the group is synchronised.

The Connector ignores Samba groups of type **5** . There is no synchronisation of SID or RID.

### Particularities

- In AD the *Pre-Windows 2000 name* (LDAP attribute *SamAccountName*) is used; therefore a group might appear under a different name in the AD configuration from that in the UCS.
- Newly created or moved groups are always created in the same sub-container on the other side. If groups of the same name exist in different containers during initialisation, then the members will be synchronised yet not their positions in LDAP. If such a group is moved on one side, the target container on the other side will be identical so that the DNs of the groups will no longer be different from this time on.
- Certain group names are realised by means of a mapping table so that the UCS group *Domain Users* for example will be synchronised with the AD group *Domänen-Benutzer* if a German language AD is used (otherwise both containers are named identically). In AD domains operating in English this feature might lead to German groups being created, so this feature should be deactivated. The Univention Configuration Registry-Variable `connector/ad/mapping/group/language` can be used for this purpose (see also chapter 6.2).

The complete table is as follows:

<i>UCS group</i>	<i>AD group</i>
Domain Users	Domänen-Benutzer
Domain Admins	Domänen-Admins
Windows Hosts	Domänencomputer

- The representation of groups in groups is different from AD to UCS. If groups are members of groups in UCS, then these objects cannot always be synchronised on the AD side; they will show up in the list of rejected objects instead. Nested groups should therefore only be assigned in Active Directory due to the existing restrictions.
- Active Directory limits the number of entries in lists of search results to a maximum of 1000 objects. During the synchronisation proper, the connector circumvents this restriction by requesting chunks of data. In certain situations the limit may nevertheless be reached. If a group was assigned as a primary group for more than 1000 users, the limit will cause a rejection of the group. This often occurs with the group **Domain Users** in large Active Directory environments. This is why the **MaxPageSize** should be extended in Active Directory (see <http://support.microsoft.com/kb/315071>).

### 5.3 Users

Users are synchronised in the same way as groups by means of their user names or their AD Pre-Windows 2000 names. The attributes **First name**, **Last name**, **Primary group** (if they exist on the other side), **Organisation**, **Description**, **Street**, **City**, **Postal code**, **Profile path**, **Login script path**, **Deactivated** and **Account expiry date** are directly transmitted. In addition, **Password**, **Password expiry date** and **Change password at next login**, are synchronised indirectly. **Primary mail address** and **Phone number** are in preparation; these attributes are commented out at present due to differences in syntax in the configuration.

The users **root** and **Administrator** are excluded.

#### Particularities

- Like groups, users are also identified by their names; thus, with regard to their position in the LDAP, the same behaviour applies to users which were created on both sides before the first synchronisation.
- On the UCS side, synchronisation of the expiry date of the password and of the password changes during next login is only carried out at the Samba level. If the password change is initiated in Univention Directory Manager yet the changes carried out against the AD, then the expiry dates related to the Kerberos password and Posix password are not changed; as a result the user has to change his password, say at the Thin Client, once more.
- During the initial synchronisation, the LDAP user **Administrator** is removed from the **Domain Admins** group since in most cases, the AD administrator is not a member of this group. As a result, the **Administrator** loses his write access to the LDAP. To avoid such a situation, an additional user can be created in the **Domain Admins** group prior to the synchronisation, which can subsequently be used for administration.
- It might be the case that a user which is to be created in AD and whose password was rejected is deleted from AD if it was created anew immediately after the password rejection. This is due to the fact that AD creates this user and immediately

deletes him after the password was rejected. If these operations are transferred to the UCS, they will be returned to AD. If the user was created anew in AD before the operation was returned, then he will be deleted after the transfer. The occurrence of such behaviour depends on the sleep period specified for the Connector.

- By default, AD and the UCS position new users in a certain primary group (usually **Domain Users**). During the initial synchronisation from the UCS to AD the users therefore inevitably become members of this group.

## 6 Univention Configuration Registry variables

A certain number of Univention Configuration Registry variables are available for configuration on the UCS side. These variables are interpreted during a restart of the connector.

### 6.1 Basic configuration

- `connector/ad/ldap/base`  
LDAP base of the Active Directory.
- `connector/ad/ldap/binddn`  
BIND DN of the account to be used in Active Directory.
- `connector/ad/ldap/bindpw`  
File (with full path) in which the password to be used against Active Directory is stored. The file should contain exactly one line.
- `connector/ad/ldap/certificate`  
File (with full path) in which the certificate exported from Active Directory is stored (for the encrypted transfer of passwords).
- `connector/ad/ldap/host`  
Fully qualified hostname of the Active Directory server.
- `connector/ad/ldap/port`  
LDAP port of the Active Directory server, preset to 389.
- `connector/ad/listener/dir`  
Directory in which the objects to be transferred from UCS to Active Directory are located; preset to `/var/lib/univention-connector/ad`. The Univention Directory Listener module stores the changes in this path; it should, therefore, not be modified.
- `connector/ad/poll/sleep`  
The time in seconds after one run without changes, before a new request is made. Locally, a search is carried out exclusively for new files in the above-mentioned directory. On the Active Directory side, an LDAP query is initiated. The lower the time set, the higher the replication speed and thus unnecessary system load accruing. The default is five seconds.

- `connector/ad/retryrejected`  
The number of queries without new changes, after which an attempt is made to input detained changes. The default is 10. This behaviour can be traced in the file `/var/log/univention/connector-status.log`.
- `connector/debug/level`  
Determines the debug information to be written to `/var/log/univention/connector.log`. Default is 1, so that warnings and errors are logged. This value can be increased up to 4.
- `connector/debug/function`  
Default is 0. If set to 1, function calls will be logged as additional debug information.

## 6.2 Predefined mapping definitions

- `connector/ad/mapping/syncmode`  
Defines the synchronisation mode; supported values are **read** (reading only from Active Directory to UCS), **write** (writing only from UCS to Active Directory) and **sync** (bi-directional synchronisation). Default is **sync**.
- `connector/ad/mapping/user/primarymail`  
Defines whether the primary mail address of UCS user objects is to be synchronised with the **mail** attribute in Active Directory. Since **mail** is a multi-value type, problems may occur during synchronisation; therefore the value is preset to **false**. During installation of the package **univention-ad-connector-exchange**, the value is set to **true**.
- `connector/ad/mapping/group/primarymail`  
Defines whether the primary mail address of UCS group objects is to be synchronised with the **mail** attribute in Active Directory. Since **mail** is a multi-value type, problems may occur during synchronisation; therefore the value is preset to **false**. During installation of the package **univention-ad-connector-exchange**, the value is set to **true**.
- `connector/ad/mapping/group/language`  
Defines which mapping of standard group names between UCS (group names are always in English) and Active Directory is to be used. Default value is mapping to a German Active Directory via the value **de**.

## 6.3 Synchronisation with Windows 2000

- `connector/ad/windows_version`  
The types of LDAP database access differ between Windows 2000 and Windows 2003. To be able to synchronise against an AD from Windows 2000, this Univention Configuration Registry variable has to be configured to **win2000**.
- `connector/ad/mapping/user/win2000/description`  
Due to internal restrictions of Windows 2000 Server, the Connector cannot create

empty object descriptions in Active Directory. Therefore, synchronisation of descriptions is deactivated in Windows 2000 mode. If this Univention Configuration Registry variable is set to **true**, descriptions for users will nevertheless be synchronised as far as possible.

## 7 Tools

The following diagnosis tools will be installed together with the Active Directory connector:

### 7.1 univention-adsearch

`univention-adsearch` allows a simple LDAP search in Active Directory. For this purpose the configuration of the AD server and AD account are taken from Univention Configuration Registry. Objects deleted in AD are still displayed (since such objects are moved to a LDAP subtree in Active Directory). As first argument, the script expects an LDAP filter, the second argument can be a list of the LDAP attributes to be displayed.

Example:

```
univention-adsearch cn=administrator cn,givenName
```

AD by default restricts the number of results to a maximum of 1000 results. If the search mask returns more entries, you will receive a corresponding error message (Size limit exceeded).

### 7.2 univention-connector-list-rejected

`univention-connector-list-rejected` lists the DNs of unsynchronised objects. In addition, the corresponding DN in the other LDAP will be displayed. Finally, **lastUSN** states the ID of the last changes that were synchronised from AD.

This script might return an error message or incomplete output if the AD Connector is currently running.

To assist troubleshooting during synchronisation, the corresponding error messages can be found in the following logfiles:

```
/var/log/univention/connector.log  
/var/log/univention/connector-status.log
```