Systems - AD: Manage users

Introduction

This tutorial will show you how to connect AD as a target system for users (their accounts) from CzechIdM. We will use an AD bundle connector from ConnId.

You can as well use newer tutorial to use wizard for AD connection - you still will need this page to explain attributes not covered by wizard and troubleshooting.

Before you start

Adding Active Directory connector

Since CzechIdM 9.2, the forked ConnId AD connector is bundled inside CzechIdM by default. You can use it out of hand to test the basic functionality. However, it is advised to use the WinRM + AD connector for the production-ready integration of CzechIdM <→ AD, as it enables more complex functionality.

Preparing Active Directory

You must prepare your Active Directory for the CzechIdM integration, mainly:

- Enable LDAPS (SSL-protected LDAP protocol) on the AD. This is vital for production deployments. Also, CzechIdM will not manage users' passwords if not connected to AD through LDAPS.
- Create an user account for the CzechIdM. Identity manager will use this account to perform operations on your AD. Although you can use a Domain Administrator account, we highly discourage it.
    - This is simply a Domain User account like any other, but you should create it in different subtree than you want to manage through IdM.
- Grant the CzechIdM user permissions on your AD.

Granting permissions

Suppose we have a domain `PISKOVISTE.BCV` with corresponding domain components `DC=piskoviste,DC=bcv` and the IdM application user is `CzechIdM (czechidm@piskoviste.bcv)`.

- CzechIdM needs to read AD configuration and schema subtrees. In our case:
    - `CN=Configuration,DC=piskoviste,DC=bcv`
    - `CN=Schema,CN=Configuration,DC=piskoviste,DC=bcv`

Ability to read schema and sufficient AD configuration should be there by default for an authenticated

user. Probably no need to adjust it.

- CzechIdM needs `full control` on subtrees which it will manage. Suppose we need to manage users, groups and computers and that we have a fairly simple setup. We grant full control to those subtrees:
  - `CN=Computers,DC=piskoviste,DC=bcv`
  - `CN=Users,DC=piskoviste,DC=bcv`
  - `OU=Groups,DC=piskoviste,DC=bcv`

Which subtrees you need to grant privileges on depends on the actual directory tree of your Active Directory.

Granting full control to CzechIdM application user

The process is fairly straightforward. Just repeat it for every root of every subtree you need to grant the rights on.

1. Open the `Active Directory Users and Computers`.
2. Right-click a container (in our case it was simply marked `Users`).
3. Choose `Delegate Control`.
4. Choose the `CzechIdM (czechidm@piskoviste.bcv)` user.
5. Choose `Create a custom task to delegate`.
6. Choose `This folder, existing objects in this folder, and creation of new objects in this folder`.
7. Tick the `Full Control` checkbox. This will tick all possible checkboxes in the dialog window.
8. Check the summary and finish the wizard. Changes are effective immediately.
9. Repeat for other subtrees as necessary.

CzechIdM has to have access to objects directly referenced from objects you manage.

For example:

A user is member of some groups, this is noted in his `member` attribute. If you want to manage the `member` attribute, the CzechIdM also has to have full access to the subtree with user groups. However this requirement is not transitive in groups hierarchy. In AD, you have a `Groups\Domain Users` group and every domain user is a member of this group. This means that every domain user has a `member` attribute which contains the `Groups\Domain Users` group DN. But the `Groups\Domain Users` is itself a member of `Builtin\Users` group.

If you want to manage your users and their group membership, you therefore need to grant full control on `Users` (to manage users) and `Groups` (because this is where `Domain Users` group is) **even if you do not want to manage groups themselves**. This is because of consistency checks performed by CzechIdM upon account provisioning.

But you **do not need** to grant anything on `Builtin` because this is referenced from an user account only indirectly.

Basic configuration

Go to **Systems** from main menu, then above list of current systems use Add button. On the first page just fill system name.

On the same page you may need to set new password policy in case that your default policy does not meet all requirements of AD configuration. If you do not want to generate passwords for AD accounts, you can skip this step at all.

Connector configuration

In next step switch to menu **Configuration** of your new system. At first, you need to chose connector, which in this case is **net.tirasa.connid.bundles.ad.ADConnector(connId)**. It will open specific configuration for that choice.

Thereafter fill important fields.

Example configuration for our local AD:

- **SSL** - this is strongly advised to enable and also vital for managing the users' password
- **Server hostname** - hostname of the AD domain controller. (IP address could be used as well, but then it must be stated in the server's certificate Subject Alternative Name)
- **Server port** - typically 636. (389 if not using SSL)
- **Failover** - an optional list of other domain controllers used in the case that the primary server is not available. Use URL format `ldaps://123.456.789.012:636`. If using multiple values, write each value at a separate line.
- **Principal** - login@domain of the user with admin privilege that CzechIdM will use for the connection. DN of the user works too.
- **Principal password** - password of the administrator user
- **Root suffixes** - the list distinguished names of the roots that connector uses for managing users. If you do not want to manage some account, it is advised not to include them in the Root suffixes. When you configure the system for the first time, root suffix should lead to the top container (e.g. DC=company,DC=local), so the system schema can be correctly generated.
- **User search scope** - manage users in specified container or subtrees. Usually subtree
- **Entry object classes** - only objects (accounts) with object classes specified there will be managed. Each object class on new line, no comma or another separator. Usual values: top, person, organizationalPerson, user.
- **Base contexts for group entry searches** - containers in AD where the groups are located. This must be specified if the groups are in different container than people and the group container is not under the path which is in "Root suffixes". You need to put it here, otherwise connector will not be able to load users' groups membership
- **Base contexts for user entry searches** - usually the same as "Root suffixes".
- **Group members reference attribute** - typically "member", use this if you want to manage group membership of user accounts
- **useVlvControls** - enable the option. (This option is only available if you use connector that is customized by BCV solutions)
- **pageSize** - we advise to set it to **100**. If you let the default 0, or ask for more than the limit for AD is, you will get an error when reading accounts. (This option is only available if you use connector that is customized by BCV solutions)

- **vlvSortAttribute** - set to "sAMAccountName"
- **Uid Attribute** - this is one of the most important option. It defines the primary key/UID of the account. Attribute values will be stored in CzechIdM for each account. Must be unique and should not change. **It is strongly advised to use "sAMAccountName", since connId connector has some problem with returning this specific attribute if mapped by other means.**
- **Object classes to synchronize** - usually the same as "Entry object classes"
- **Specified attributes to be returned** - default "ldapGroups" and "sAMAccountName". This option is also used when you need to read attributes from AD, that are not returned by default, a typical example is extensionAttribute1 and other additional attributes.

If you are setting this on a Windows server, make sure to delete the 'Specified attributes to be returned' values and write them manually. Otherwise, ldapGroups will not be returned due to some white space problems

Beware on **useVlvControls** option. CzechIdM now only supports vlv control, so **useVlvControls** option should be enabled and **vlvSortAttribute** must be set (recommended option - 'sAMAccountName'). **DO NOT** use **CN**, **distinguishedName** or any other unindexed attribute or you'll end up with "[LDAP: error code 12 - 0000217A: SvcErr: DSID-03140414, problem 5010 (UNAVAIL_EXTENSION), data 0];" error

Since connector version 1.3.4.25 we support change of **sAMAccountName**, even if it is used as identifier (in provisioning mapping use sAMAccountName instead of __Uid__)

Since connector version 1.3.4.25 we support objectGUID as identifier, but only with this property turned off:

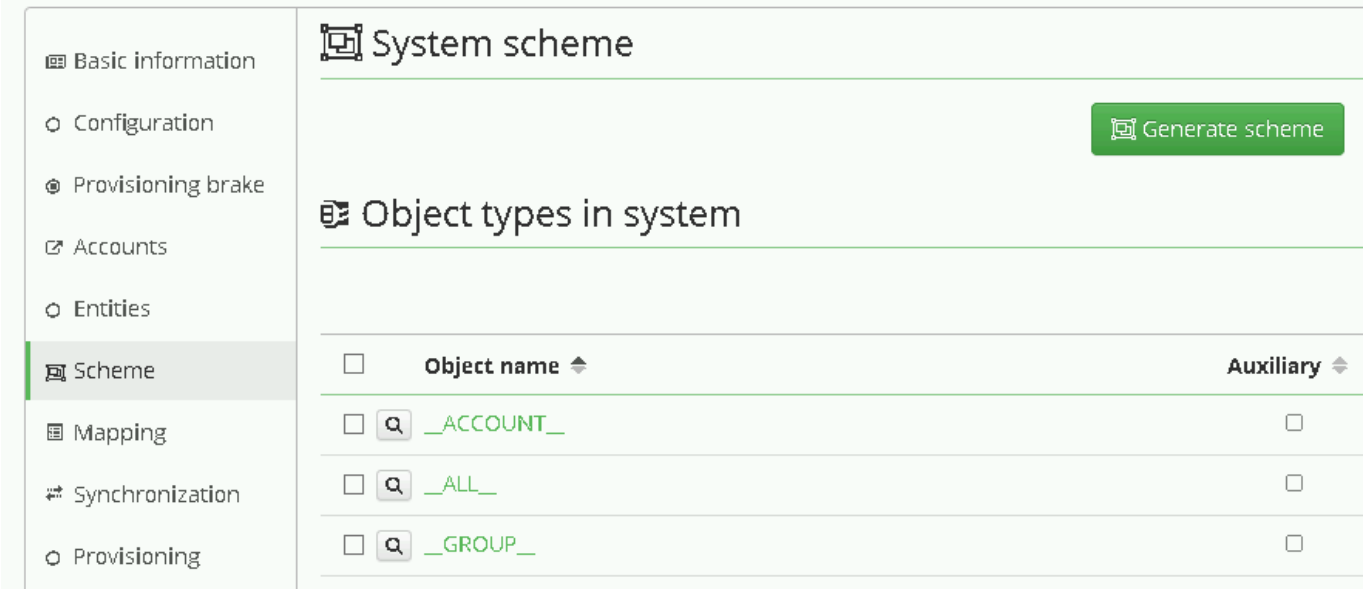
```
idm.sec.acc.provisioning.allowedAutoMappingOnExistingAccount=false
```

(in create action you have to send random String, because UID cannot be null and objectGUID will be obtained after create of user/group)

Scheme

For next step, go to menu **Scheme** on your system.

You can let CzechIdM generate a scheme for you by clicking on **Generate scheme** button and that is also the preferred way. For MS AD, the connector usually creates 3 object types. \ACCOUNTS\, \ALL\, \GROUP\.

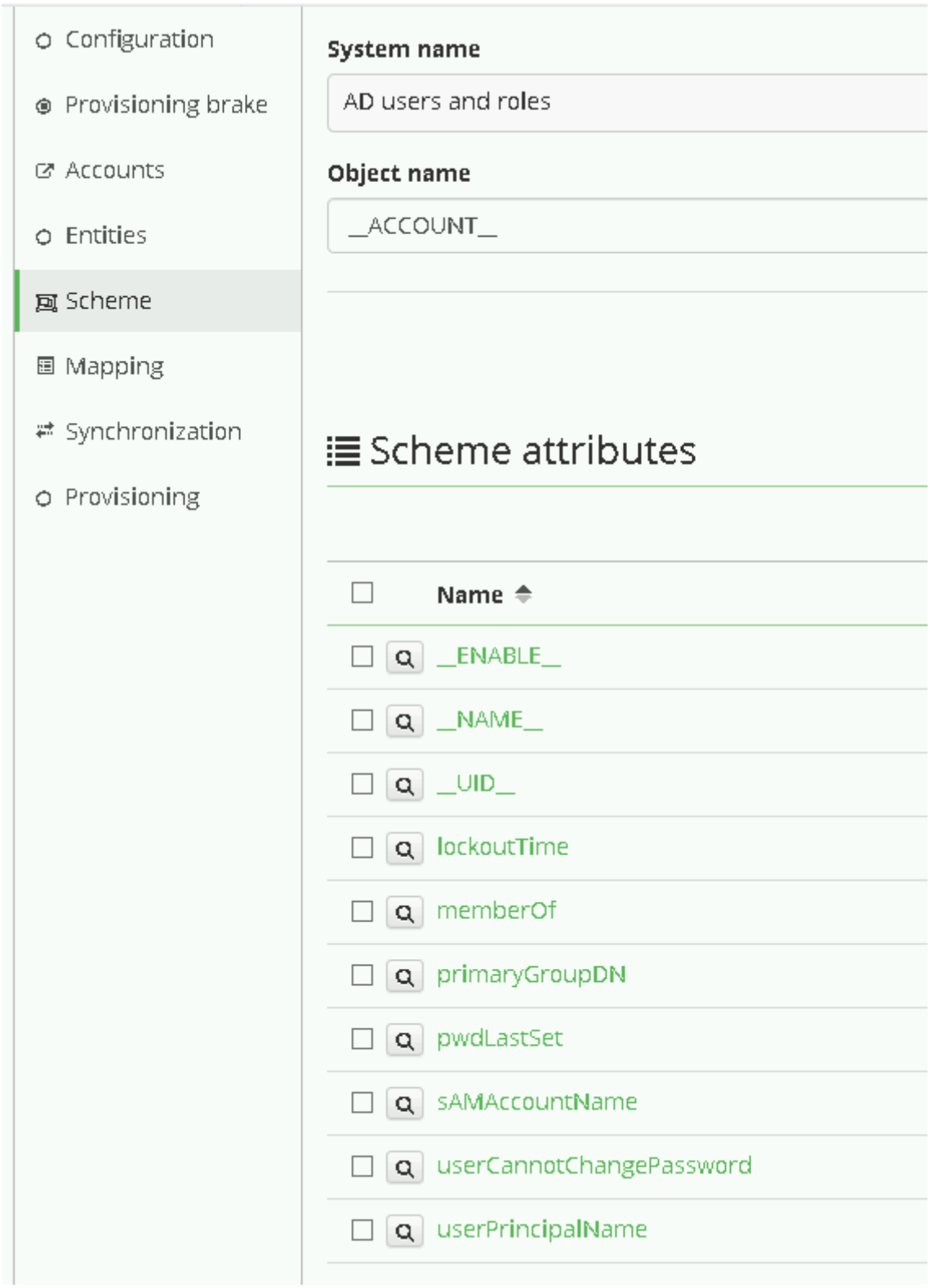


For user management, we will use __ACCOUNT__. Click on the detail of the object type and check that the scheme attributes list consists of all attributes you want to manage in AD. If the list doesn't contain any attribute or contains 6 or less, check that **Root suffixes** in the system configuration contains the value of the top container (so the connector can read the schema definitions).

If you are connecting AD for the first time, it is a good idea to check some minimal set of attributes that allows you to create an account, which is usually:

- sAMAccountName - this attribute is sometimes not generated by default (mainly if it isn't used as Uid). If so, you must create it manually. Use the button **Add**, fill in the name "sAMAccountName", type "java.lang.String", able to read, update, create and returned by default.
- __ENABLE__ - if you want to allow disabling a user in AD. This attribute is not generated by default, so you can create it manually. Use the button **Add**, fill in the name "__ENABLE_\ckgedit>, type "java.lang.Boolean", able to read, update, create and returned by default. * __NAME__ (synonymous to DN, hard-coded in the connector). This attribute should be generated by default. If not, use the button **Add**, fill in the name "__NAME_\ckgedit>, type "java.lang.String", able to read, update, create and returned by default.
- __PASSWORD__ - this special attribute is used for setting the passwords for user accounts. User in AD can't be activated when a password is not set. This attribute is not created by default in the schema, so you must add it manually: name "__PASSWORD__", type "eu.bcvsolutions.idm.core.security.api.domain.GuardedString", able to update, create
- ldapGroups - use this attribute if you want to manage users' group membership. This attribute is not created by default, add it manually: name "ldapGroups", type "java.lang.String", able to read, multivalued, able to create, edit, returned by default


You do not need to use all of the schema attributes for provisioning afterwards



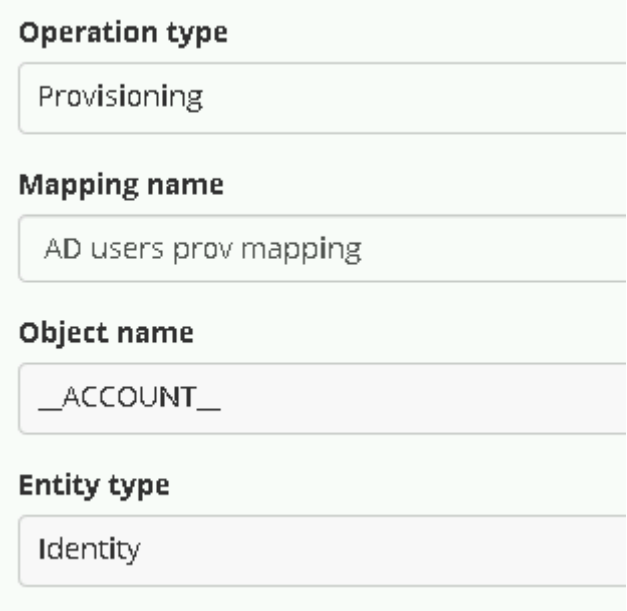
It is possible you will not see the full scheme even with root suffix set to the top container. In that case, check that schemas are not stored separately and if they are, set root suffixes to the appropriate DC.

Mapping

Now go to menu **Mapping**. There you must set, which attribute from scheme is mapped to which attribute in CzechIdM.

At first set:

- **Operation type:** Provisioning - we want to manage data in AD from CzechIdM
- **Object name:** __ACCOUNT__ - this is a standard type of scheme object in AD
- **Entity type:** Identity - this entity type in CzechIdM we want to provision
- As **Mapping name** set whatever you want, for example **AD users prov mapping**.



Then map scheme attributes to entity attributes as described below: **sAMAccountName**

- Name - sAMAccountName
- Identifier - true
- Entity attribute - true. Means that the attribute will be filled from basic Identity attribute set.
- Entity field (selectbox) - User name

It is strongly advised to use **sAMAccountName** as an identifier, so that the identifier of the connector is the same as the identifier of the provisioning. Then some advanced CzechIdM features can be used and the debugging is also much easier

Other options may stay with default values.

__ENABLE__, mapping configuration is almost the same as **sAMAccountName**, but do not set it as identifier. Map this schema attribute to entity attribute "Disabled". You should also add transformation to the system, because CzechIdM holds the attribute "disabled" and AD has attribute "enable". So the transformation should return opposite value of the attribute in CzechIdM. To do so, click on the **Insert script** button in "Transformation to system" window and find the script **getOppositeBoolean**. This will fill the window with the script call, but you must also add the line `.addParameter('attributeValue', attributeValue)` after the similar line with

"scriptEvaluator" (see this tutorial for using Standard transformation scripts).

If you also want to create entities in AD, which is probable, map **__NAME__** attribute that holds the DN of the account in AD. The configuration of the attribute may look like:

- Attribute in schema - __NAME__,
- Name - DN(__NAME__)
- Entity attribute - true
- Entity field - user name. In case that the DN on AD consists of the login of the user. Otherwise, you should choose other attribute or EAV.
- The form of the DN varies on each instance of AD, so there usually will be some **transformation to system** like `return "CN=" + attributeValue + ",OU=employees,DC=yourcompany,DC=com"` . Of course your tree can be more complex, in that case you should follow some of our tutorials

Password mapping

If you want to send passwords into Active Directory, you need to configure SSL communication.

To enable passwords provisioning, add the attribute **__PASSWORD__** to the schema attributes (as written above) and map it as follows:

- Attribute in schema - __PASSWORD__,
- Name - __PASSWORD__
- Entity attribute - false
- Attribute with password - true

Forced password change (User must change password at next logon)

When mapping AD attributes, it is sometimes useful to be able to set a forced password change option. This requirement is often set for two different cases:

* We need to change the password when logging into AD **for a new user account** * We need to force a password change but **only after a password reset**

1/ To force a password change for newly created users, map the **"pwdLastSet"** attribute. The attribute should be in the generated system schema, object "**__ACCOUNT__**" name "pwdLastSet", Data type "java.lang.Boolean". So add the attribute to the mapping and put "return true" in the transformation script(Transformation to system) and set the strategy "Write only on create of the entity".

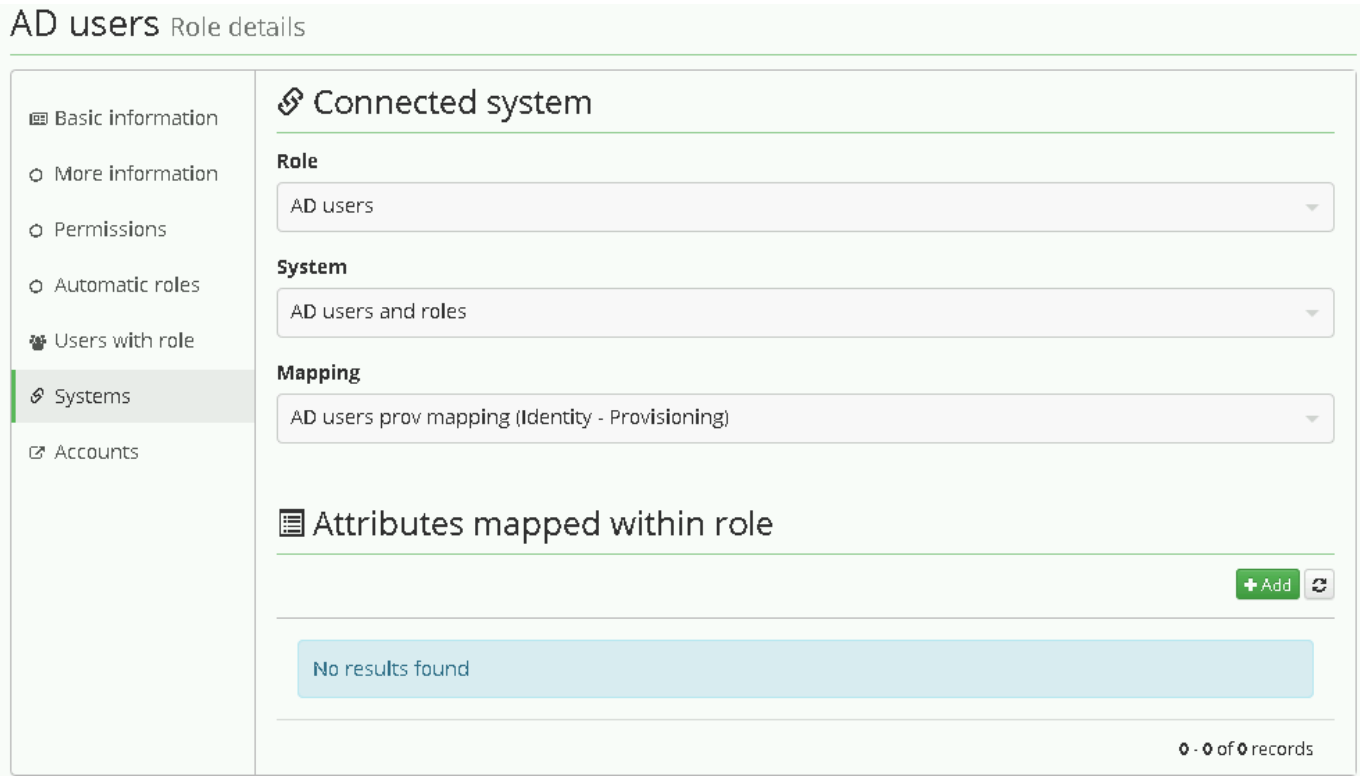
2/ If we need to force password change every time password is reset, map attribute pwdLastSet too, but **with checkbox "Include on password" and "Include only when password is changed"** and strategy "Set value as it is". This can only be set since IdM version 11.0. In the picture you can see the attribute in the active directory.

Role for AD

When the mapping is set, the last step is to define a role in CzechIdM, that grants the user the account in AD. Prepare a new role in CzechIdM only with basic attributes. Name should be sufficient.

If you want some more options, follow How to create a role.

Then in the role detail, go the the menu tab **Systems**, add a new system and choose previously created one - "AD users and roles". Then also choose your provisioning mapping, there should be only one, and save it.



From now on, every time user gets the role, it is provisioned into the connected system AD. You can see that on users detail menu tab "Provisioning" or in the audit "Provisioning"

Check if the user created in IdM has password according to Active directory password policy, otherwise user can be created in IdM but not provisioned to AD system.

Group membership

If you want to add a user to an AD group via CzechIdM, you need appropriate Role there. So create a role that is almost similar to **AD - Users** above. Give it some better name that represents the group in AD, good idea is to use sAMAccountName or CN e.g. **CRM basic user**.

In the system AD - users schema setting, you need to have a special attribute defined that add the user to the group. The attribute is named **ldapGroups**. Make sure, it can be read,create,update,delete. Attribute is multivalued. In the system configuration tab, there are some configuration properties, that must be set in order to allow group membership management.

- **Group members reference attribute** - usually **member**. This represents the name of the attribute in AD that is present in Group. Its value is usually a DN of the user in the group.

Then continue to AD - users Mappings and edit provisioning mapping. Add there a **ldapGroups** attribute. It is not filled from any identity attribute and has no transformation. (It will be filled from the role). Since the attribute is multivalued, its filling strategy must be either **MERGE** (recommended for AD) or AUTHORITATIVE MERGE (info about strategies).

Get back to your role CRM basic user. In the tab **Systems** add a system **AD - users and roles**, save

it. Then add an attribute that will be filled by this role - **ldapGroups**. Again choose the filling strategy **MERGE** (or AUTH.MERGE, make sure to use the same as in the provisioning mapping). Then **add a transformation** that is the value of DN of the group in AD ' " ' sign on each side of the text.

Thus every user that has the role assigned is added to the group with provided DN via ldapGroups attribute.

For managing group membership in multi domain AD environment follow this tutorial

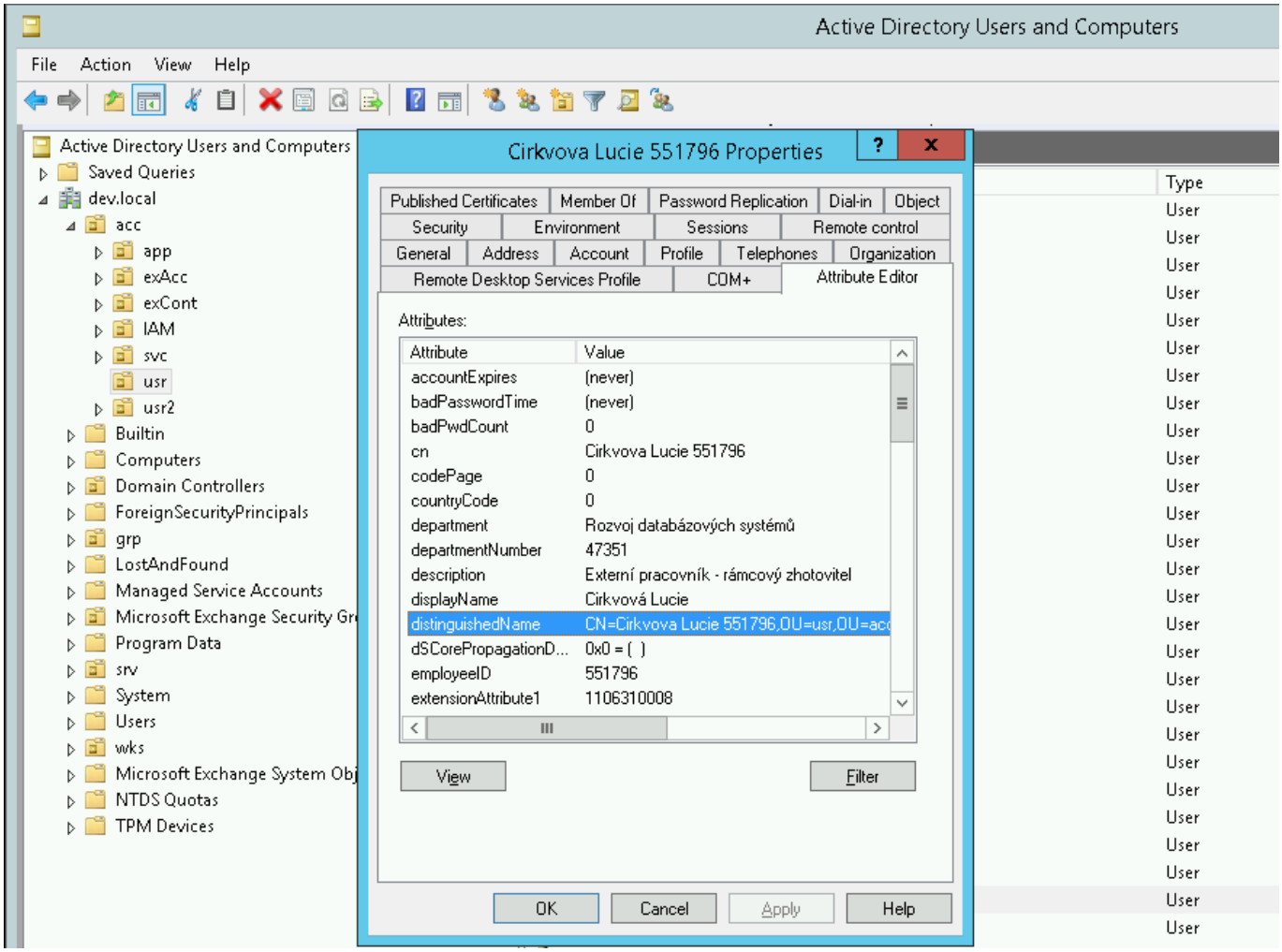
Merge was fixed in connector version 1.3.4.25. Before Merge behaved like Authoritative Merge

Tips & Tricks

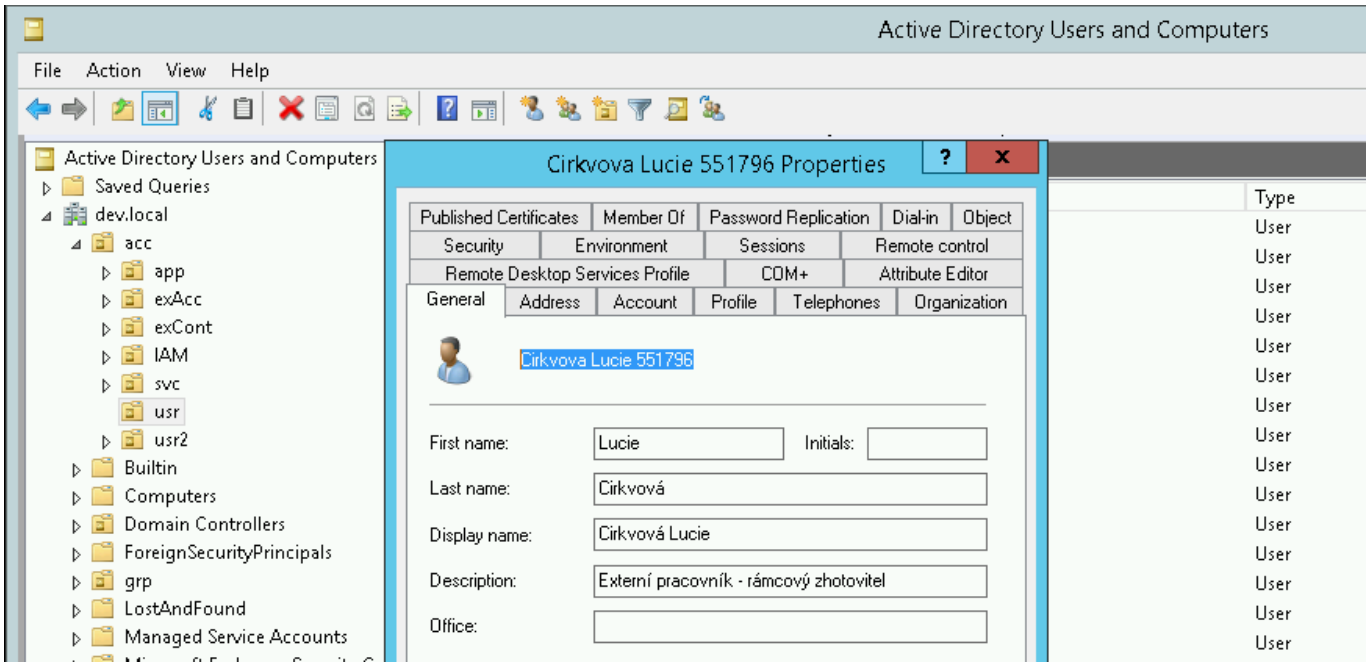
Distinguished Name (DN), Common Name (CN)

You can easily find DN of a user account with the help of **Active Directory Users and Computers** in your Windows server. Open the user's detail and switch to the tab **Attribute Editor**. You can see here the attributes in the same format as IdM sees them.

If you do not see the Attributes editor, you have to switch it on. Go to `Active Directory Users and Commputers` → `View` and select `Advanced features`.



Moreover, CN of the user account is the same as **Name** so you can see it on the first page of the user's detail next to the icon.



ldapGroups not returned

If you are running on a Windows server, the 'ldapGroups' in 'Specified attributes to be returned' has the wrong value 'ldapGroups\r' (this is only visible in Audit). The solution is to remove the value in 'Specified attributes to be returned' and write it again manually.

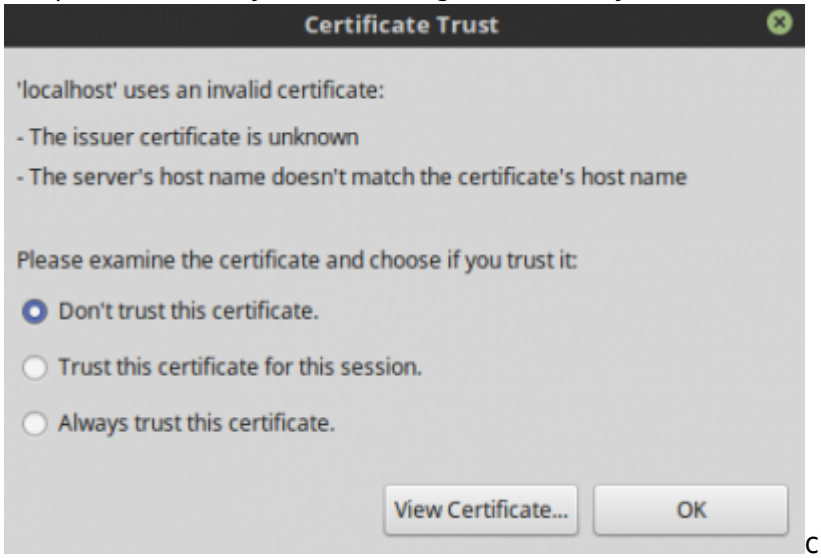
Mapping extensionAttributes

AD enables additional attributes named extensionAttribute1 - extensionAttribute10. If you want to fill these attributes by IdM, you must do following steps in the configuration of the connected system:

- Go to **Configuration** → **Specified attributes to be returned (multi)**, add **extensionAttribute1** to a new line under existing values.
- Go to **Scheme** → **__ACCOUNT__** → use the button **Add**, fill in the name **extensionAttribute1**, type "java.lang.String", select able to read, update, create and returned by default.
- Go to **Mapping** → **Provisioning mapping** → use the button **Add** and map the attribute according to your choice. The following example can be used when you want to fill the extensionAttribute1 by personal numbers of identities
  - Attribute in schema - extensionAttribute1
  - Name - extensionAttribute1
  - Entity attribute - true
  - Entity field - Personal number

Connection via SSL not working

If you just imported root certificate to IdM truststore, but SSL connection to AD is still not working try following method to find which server hostname you should use. Configure connection via SSL to AD in Apache Directory Studio during connection you will see this window:



click on View certificate → tab General → field Issued To → Common name(CN) and use this value as server hostname.

LdapErr: DSID-0C0907C5

If you see this error when reconciliating AD groups:

```
org.identityconnectors.framework.common.exceptions.ConnectorException:
javax.naming.OperationNotSupportedException: [LDAP: error code 12 -
00000057: LdapErr: DSID-0C0907C5, comment: Error processing control, data 0,
v1db1]; remaining name 'OU=company,DC=domain,DC=tld'
```

the likely cause is that some groups have many members. AD has a property MaxPageSize which is probably set to lower than necessary (default is 1000). Increasing the value to an arbitrary large number (30000) helped in our case but only AD admin can change this.

SvcErr: DSID-031007E5 - unsupported special characters in DN

The AD connector doesn't support containers (OU) that contain some special characters in their names, namely the forward slash ("/"). When you try to update some user, whose DN contains this character (even when updating only some other attributes), then the provisioning fails with the following exception:

```
javax.naming.NamingException: [LDAP: error code 1 - 000020D6: SvcErr:
DSID-031007E5, problem 5012 (DIR_ERROR), data 0
```

Please rename your containers so they don't contain special characters.

See more about this known issue here: https://redmine.czechidm.com/issues/2294.

Failover

The configuration property Failover is used when the primary server (configured in the Server hostname) is unavailable. The attribute contains a list of AD servers that connector can use.

Please note that this property is not used in the case that the primary server is accessible on the given port, but there is some other problem with the communication (e.g. the credentials are incorrect).

The value of this property must be a proper URL, e.g. `ldaps://some.hostname:636`. If using multiple values, write each value at a separate line.

Video Guide

How to create role for AD group - czech language