Active Directory with LDAP connector

Status

Provisioning works well.

Synchronization works well.

> ⓘ This connector is the recommended way to connect to the Active Directory since connector version 1.4.2.14 (bundled with midPoint 3.3.1 and 3.4).
>
> The .NET-based Active Directory connector is deprecated and it is no longer supported.

Description

The connector can be used for provisioning and synchronization with Active Directory using the LDAP protocol.

Resource Configuration

(Remote connector server is not needed for this connector)

Administrative Account for Provisioning/Synchronization

We have successfully tested both provisioning and synchronization of users with the following access privileges using Active Directory domain "Delegate Control" mechanism:

- Create, delete and manage user accounts
- Reset user passwords and force password change at next logon
- Read all user information
- Create, delete and manage groups
- Modify the membership of a group
- Create, delete and manage inetOrgPerson accounts (TODO: is this needed?)
- Reset inetOrgPerson accounts and force password change at next logon (TODO: is this needed?)
- Read all inetOrgPerson information

Recommended Connector

Framework: ConnId
Bundle: com.evolveum.polygon.connector-ldap
Version: most recent stable version
Connector: com.evolveum.polygon.connector.ldap.ad.AdLdapConnector

Connector Configuration

(currently, no published documentation)

Active Directory LDAP Strangeness

Active Directory in the default configuration is not really LDAPv3-compliant server. It has many quirks, extensions, modification and twists the LDAP standard almost beyond recognition. The LDAP connector was modified to survive this brutal "intepretation" of the LDAP specifications. However, there are many things that needs to be taken into account when configuring AD resource:

- `instanceType`, `nTSecurityDescriptor` and `objectCategory` are formally defined as mandatory attributes in the `top` object class (!!!). This means they are (formally) mandatory for all objects accessed using LDAP connection. But the reality is different. It seems to be OK to create an object without these attributes. Therefore for a proper operation of midPoint we recommend to modify the schema using the `limitations` mechanism in midPoint Resource Schema Handling by setting `minOccurs=0`. (This is already done in the sample referenced below.)
- The objects can easily have attributes that are not defined in any object classes that they have. E.g. a normal user (the `user` object class) may have attribute `info`. If such extra attributes are used in your AD instance then the best way is to configure them as operational attributes in the connector configuration and define them explicitly in Resource Schema Handling (see **MID-3379** - Getting issue details... STATUS).

Resource Configuration Example

```
<connectorConfiguration xmlns:icfc="http://midpoint.evolveum.com/xml/ns/public/connector/icf-1/connector-schema-3">
        <icfc:configurationProperties xmlns:icfcldap="http://midpoint.evolveum.com/xml/ns/public/connector/icf-1/bundle/com.evolveum.polygon.connector-ldap/com.evolveum.polygon.connector.ldap.ad.AdLdapConnector">
            <icfcldap:host>ad.example.com</icfcldap:host>
            <icfcldap:port>636</icfcldap:port>
            <icfcldap:baseContext>DC=evolveum,DC=com</icfcldap:baseContext>
            <icfcldap:bindDn>CN=midpoint,CN=Users,DC=evolveum,DC=com</icfcldap:bindDn>
            <icfcldap:connectionSecurity>ssl</icfcldap:connectionSecurity>
            <icfcldap:bindPassword>
                <t:clearValue>secret</t:clearValue>
            </icfcldap:bindPassword>
        </icfc:configurationProperties>
        <icfc:resultsHandlerConfiguration>
            <icfc:enableNormalizingResultsHandler>false</icfc:enableNormalizingResultsHandler>
            <icfc:enableFilteredResultsHandler>false</icfc:enableFilteredResultsHandler>
            <icfc:enableAttributesToGetSearchResultsHandler>false</icfc:enableAttributesToGetSearchResultsHandler>
        </icfc:resultsHandlerConfiguration>
    </connectorConfiguration>
```

Resource Sample

See resource sample.

Limitations

- Active Directory LDAP schema is violating LDAP standards and best practices in numerous ways. The connector is build to tolerate these "quirks" in the AD schema. However the underlying LDAP library may complain about the schema issues. It is usually safe to ignore these warnings.

Notes

Note: to avoid clear-text password visible in the repository, please refer to String to ProtectedString Connector Configuration.

Full Active Directory Schema

Active Directory has huge schema. The schema when encoded in XSD has several megabytes. This might take several hundreds of megabytes of memory when processed. Make sure that your midpoint instance has enough memory (heap) to handle that. The impact of AD schema can be limited by reducing the number of object classes that are processed by midPoint:

```
    <schema>
       <generationConstraints>
            <generateObjectClass>ri:user</generateObjectClass>
            <generateObjectClass>ri:group</generateObjectClass>
        </generationConstraints>
    </schema>
```

See also **MID-2716** - Getting issue details... STATUS

See Also

- Active Directory Connector (LDAP)
- Active Directory Tips&Tricks
- Active Directory Multi-Domain
- Active Directory with the legacy .NET connector
- AD Connector Design Notes

External links

- What is midPoint Open Source Identity & Access Management
- Evolveum - Team of IAM professionals who developed midPoint
- WILL_NOT_PERFORM - wiki page explaining a lot of error messages returned by Active Directory