How to configure a VDP database with LDAP authentication

Revision 20220208

Denodo Inc.: 525 University Avenue, Suite 31, Palo Alto, CA 94301. USA www.denodo.com

CONTENTS

1 GOAL

This document describes how to configure an LDAP database in Virtual DataPort and how to debug and troubleshoot this configuration.

It is strongly recommended to read the sections User and access rights in Virtual DataPort and Administration of databases, users, roles and their access rights of the **Virtual DataPort Administration Guide** before reading this document.

2 INTRODUCTION

Virtual DataPort can connect to LDAP directories (such as Microsoft Windows Server Active Directory) to extract data stored in the server or to delegate the authentication to the LDAP.

In addition to the technical considerations about LDAP and role authentication, there are some aspects to take into account regarding the use of this authentication type. When using LDAP authentication to connect to a database, a latency for the authentication is added.

The recommendation to avoid a big overload due to the LDAP authentication is to configure the clients to use Connection Pools against VDP. This will ensure that not all the queries need to create a new connection and authenticate against the LDAP server. For example, using JDBC access it is possible to use a connection pool to avoid the overhead.

The Lightweight Directory Access Protocol is an open, industry standard application protocol for accessing and maintaining distributed directory information services over an IP network.

Typical URL's to connect to LDAP servers are:

- `ldap://acme.server`
- `ldap://acme.server/dc=example,dc=com` (refers to all entries of the example.com directory)
- `ldaps://secured.acme.server` (LDAP over SSL)

Directories have the following model:

- A directory is composed of entries.
- An entry consists of a set of attributes (defined in a schema).

- An attribute has a name and one or more values.
- Each entry has a unique identifier: its Distinguished Name (DN).

For example, the entry for the user “John Smith” can be:

Distinguished Name (DN): `cn=John Smith,dc=example,dc=com`

Attributes:

```
givenName: John
sn: Smith
mail: john.smith@example.com
memberOF: cn=Users,dc=example,dc=com
objectClass: person
```

2.1 ACTIVE DIRECTORY

Microsoft Active Directory is based on standard directory access protocols, such as Lightweight Directory Access Protocol (LDAP). So Virtual DataPort is able to communicate to this kind of directories using the LDAP protocol.

Most typical Active Directory configurations are domains and forests. For more detailed information about the architecture of an Active Directory installation, please read https://technet.microsoft.com/en-us/library/cc961765.aspx.

2.1.1 Active Directory Domain

Domains are container objects. Domains are a collection of administratively defined objects that share a common directory database, security policies, and trust relationships with other domains. In this way, each domain is an administrative boundary for objects.

A single domain can span multiple physical locations or sites and can contain millions of objects.

To connect to a single domain, usually you can use the ldap protocol connecting to the default port 389:

- `ldap://acme.server:389`

2.1.2 Active Directory Forest

A forest is a complete instance of Active Directory. Each forest acts as a top-level container in that it houses all domain containers for that particular Active Directory instance.

So, a forest can contain one or more domain container objects, all of which share a common logical structure, **global catalog (GC)**, directory schema, and directory configuration, as well as automatic two-way transitive trust relationships.

By default, information in Active Directory is shared only within the forest. In this way, the forest is a security boundary for the information that is contained in that instance of Active Directory.

To connect to a forest, in order to delegate the authentication to the final domain, you have to connect to the GC. In most configurations, the GC can be accessed using the port 3268.

- `ldap://acme.server:3268`

3 DENODO VIRTUAL DATAPORT CONFIGURATION

The recommended steps to connect to a VDP database using LDAP authentication are:

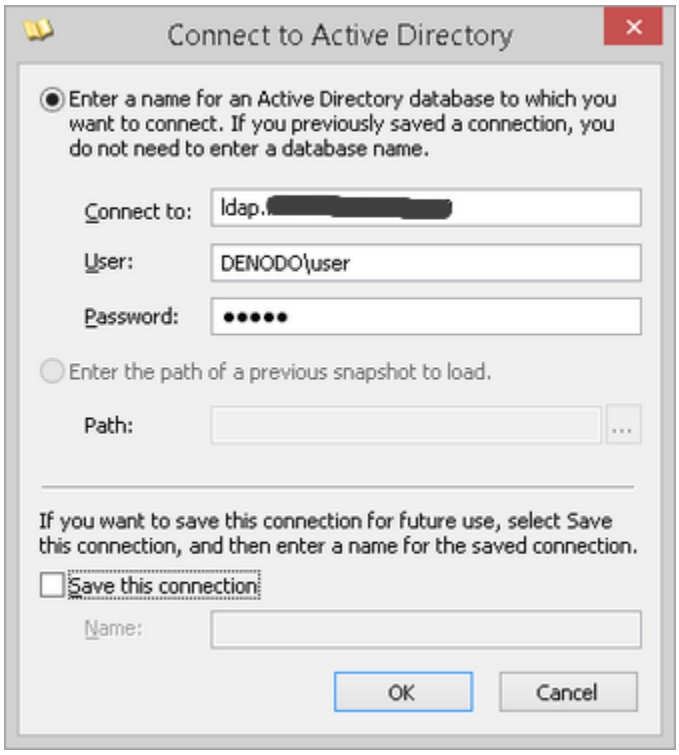
1. Check the LDAP or Active Directory configuration
2. Define an LDAP data source in VDP.
3. Enable and Configure LDAP Authentication for the server.
4. Import roles from the LDAP server.
5. Assign privileges to the LDAP roles to connect to the database.

3.1 CHECK LDAP OR ACTIVE DIRECTORY CONFIGURATION

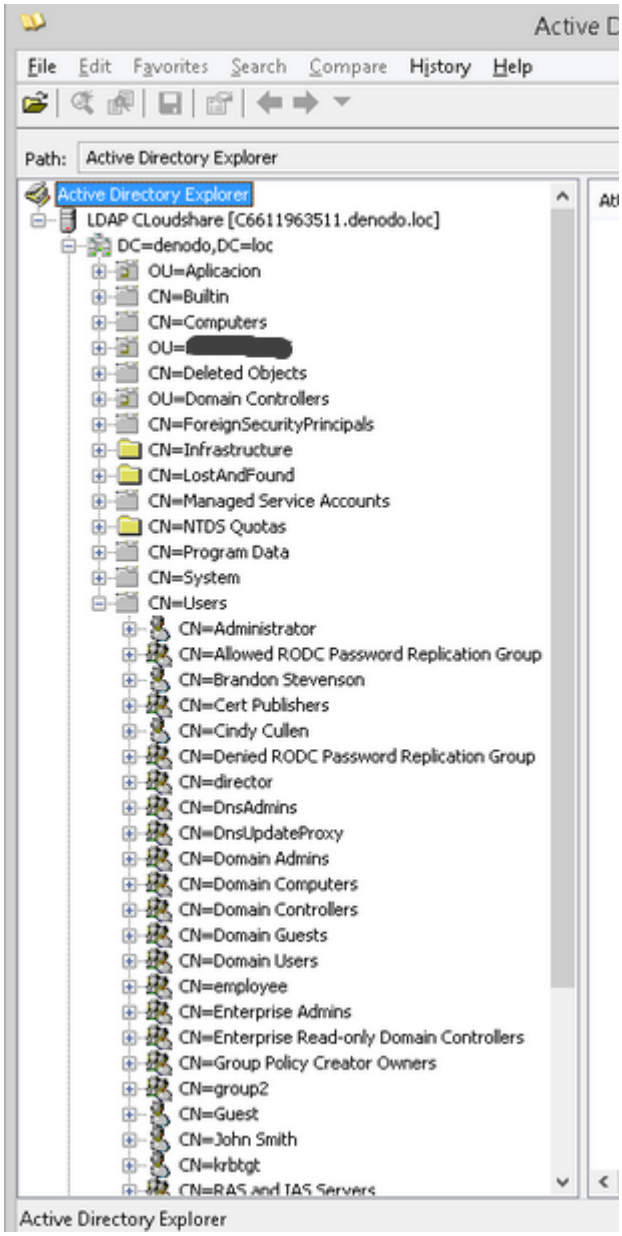
The most common scenario is to have an Active Directory with the information about users and groups (for example, departments). The first thing to do before activating the LDAP configuration in VDP is to take a look at the LDAP server itself to see how it is structured.

For this purpose, we recommend using a client application like Microsoft ADexplorer (https://docs.microsoft.com/en-us/sysinternals/downloads/adexplorer).

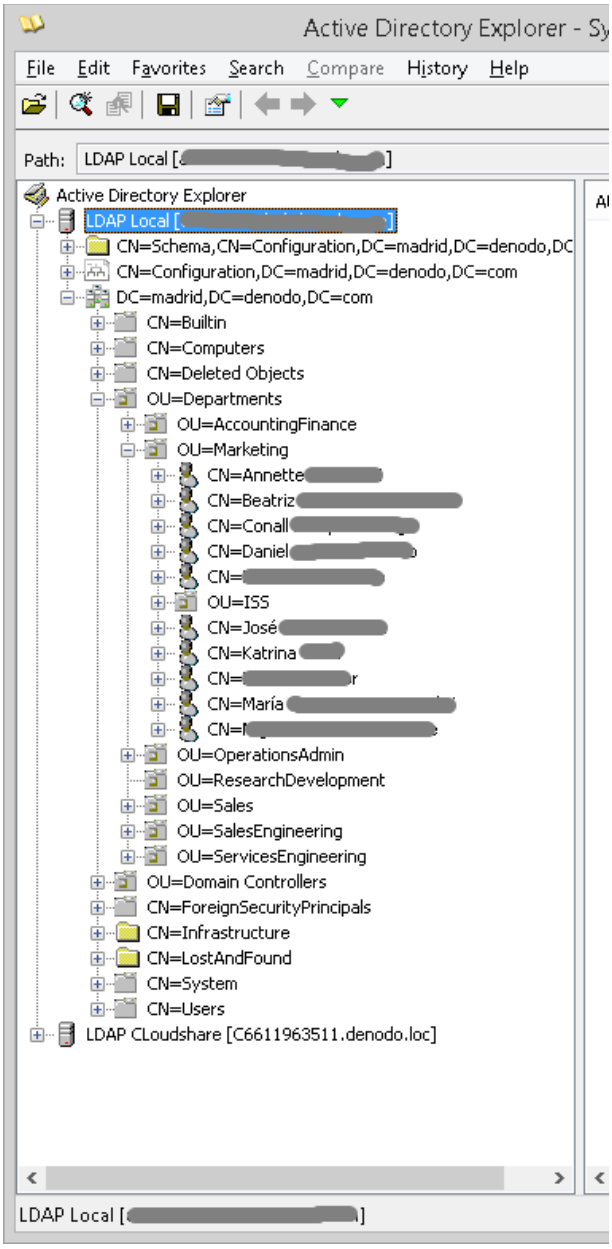
The application will prompt the user for a valid username to connect to the AD (this user needs enough privileges to search into the active directory):



When connected, the application displays in the left panel, all available nodes (grouped in folders or containers):



The image above shows a typical configuration: all users and groups are stored inside **CN=Users** container, but we can find more complex configurations, for example:

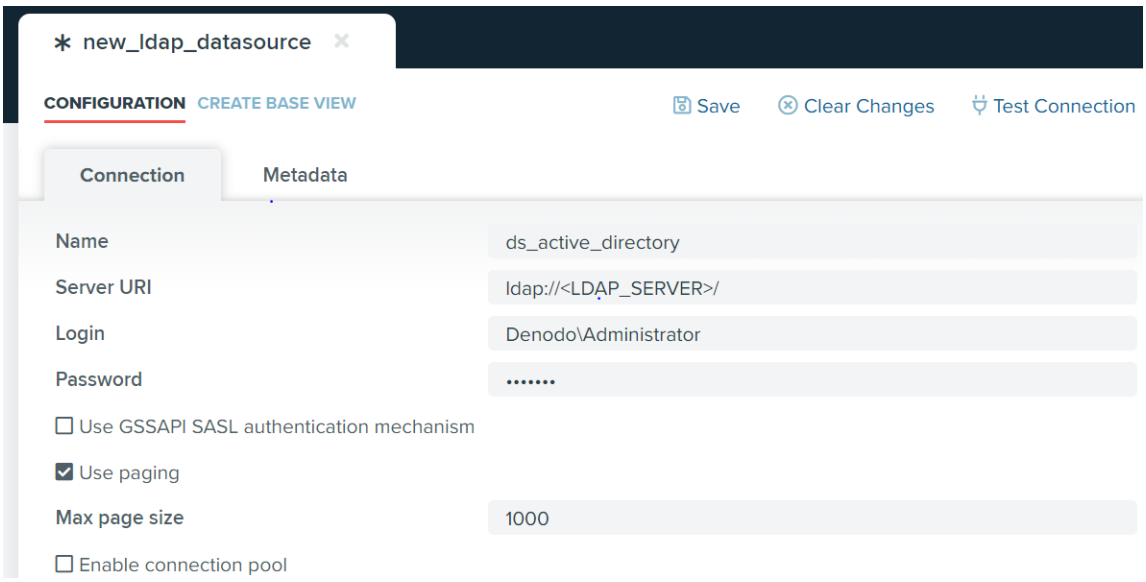


In this case, users are inside containers, and the containers (which are not groups) are the departments.

Summarizing, we have a working user to connect to the LDAP server so it's time to go to Virtual DataPort to configure the LDAP data source.

3.2 DEFINE AN LDAP DATA SOURCE IN VDP

Connect to VDP using Design Studio, for example to the `admin` database (usually this database can be used to hold the common data sources) and create a new LDAP data source (`File > New > Data source > LDAP`).



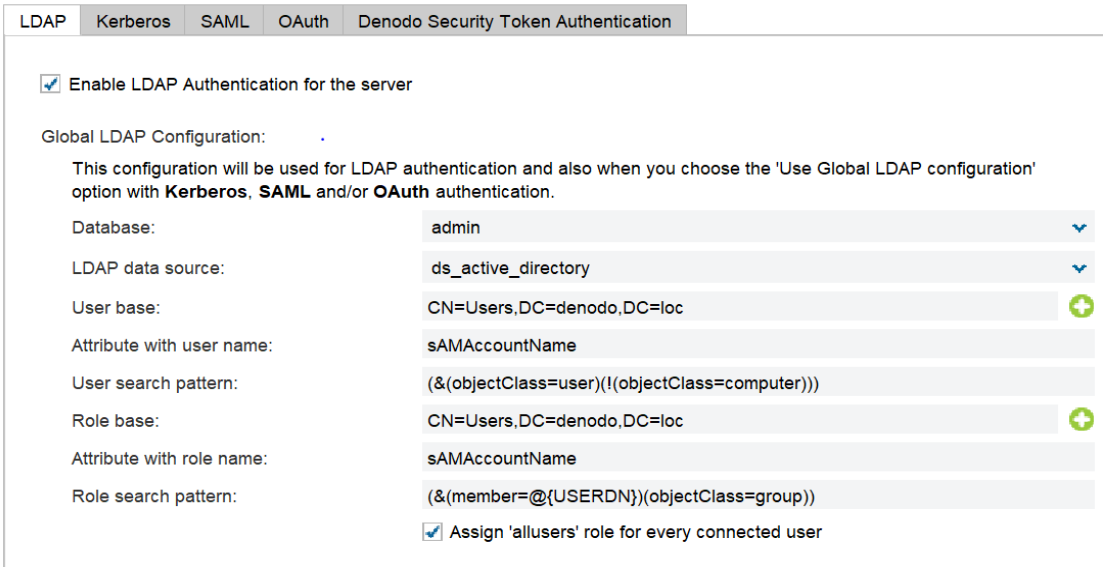
The **Server URI** and the credentials (**Login** and **Password**) to access the LDAP data source have to be provided (they are the same as the credentials used in ADexplorer in the previous step).

Note: The only difference resides in the Server URI. In VDP it has to start with "`ldap://`" or "`ldaps://`" (in ADexplorer you do not need to include the protocol).

It's recommended to check the "`Use paging`" option in case your organization has a lot of nodes (in order to retrieve all of them in blocks).

3.3 ENABLE AND CONFIGURE LDAP AUTHENTICATION FOR THE SERVER

To make Global LDAP configuration for the server, connect to VDP using the Virtual DataPort Administration Tool and navigate to
(`Administration > Server Configuration > Server Authentication > LDAP`)



Enable the check box “Enable LDAP Authentication for the server” to allow all virtual databases to be authenticated based on the Global LDAP configuration. Leaving this checkbox unchecked will allow administrators to configure and enable LDAP authentication at Database level.

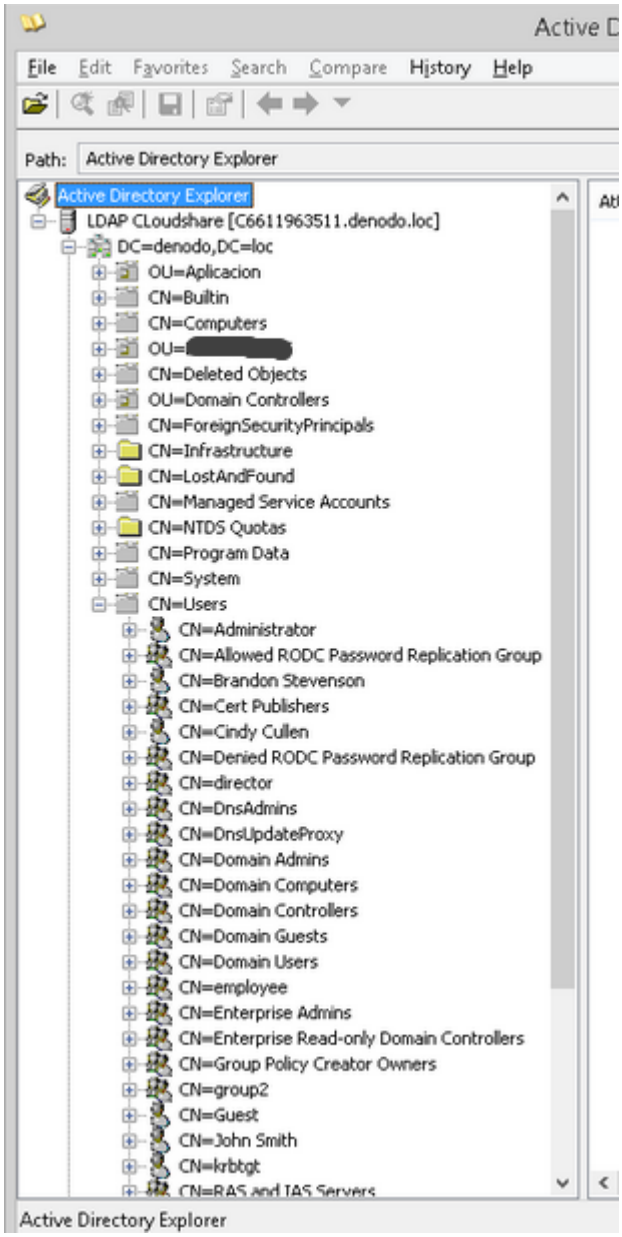
Note: For older versions prior to Denodo 8.0, LDAP Authentication can be only configured at Database level.

In order to find the user and role search patterns, we recommend the use of any third party LDAP client like ADExplorer or JXplorer or Apache Directory studio to execute and test the queries to the LDAP server if needed.

The information that is required from the LDAP administrator is:

- **User base**: node of the LDAP server that is used as scope to search nodes that represent users. It is possible to specify more than one.
- **Attribute with user name**: name of the attribute that will be used as login identifier.
- **User search pattern**: pattern used to retrieve the users from the LDAP.
- **Role base**: node of the LDAP server that is used as scope to search nodes that represent roles. It is possible to specify more than one. In some basic configurations it is the same as the **User base**.
- **Attribute with role name**: name of the attribute that contains the name of the role.
- **Role search pattern**: pattern used to generate the LDAP queries that will be executed to obtain the nodes that represent the roles of a user. This pattern has to contain the token `@{USERDN}`, which will be replaced with the Distinguished Name of the user that tries to connect to the database.

Let’s see an example: we are going to use the first AD configuration we saw at the beginning of this document. The configuration was:



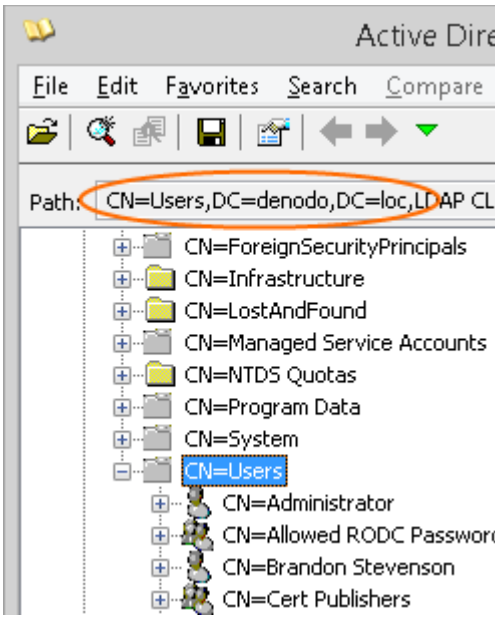
3.3.1 User base

In the image above, you can see the users stored inside `CN=Users`:

- `CN=Administrator`
- `CN=Brandon Stevenson`
- `CN=Cindy Cullen`

- `CN=Guest`
- `CN=John Smith`
- …

As all our users are in the same container, our **User base** parameter will be only one node, but in the configuration we have to put the complete distinguished name. Selecting the element in ADexplorer will show the complete value (`CN=Users,DC=denodo,DC=loc`):



3.3.2 Attribute with user name

The next parameter is the **Attribute with user name** so now we have to select the attribute used for the VDP user name at login time. Let's select a user and see all his attributes:

Attribute	Syntax	Count	Value(s)
accountExpires	Integer8	1	0x7FFFFFFFFFFFFFFF
badPasswordTime	Integer8	1	7/28/2014 8:07:00 PM
badPwdCount	Integer	1	1
cn	DirectoryString	1	Brandon Stevenson
codePage	Integer	1	0
countryCode	Integer	1	0
displayName	DirectoryString	1	Brandon Stevenson
distinguishedName	DN	1	CN=Brandon Stevenson,CN=Users,DC=denodo,DC=loc
dSCorePropagationData	GeneralizedTime	4	2/11/2014 3:17:35 AM;12/13/2013 2:24:26 PM;4/2/2013 5:5
givenName	DirectoryString	1	Brandon Stevenson
instanceType	Integer	1	4
lastLogoff	Integer8	1	0x0
lastLogon	Integer8	1	6/19/2014 2:31:10 PM
lastLogonTimestamp	Integer8	1	10/22/2014 4:21:33 PM
logonCount	Integer	1	0
memberOf	DN	1	CN=employee,CN=Users,DC=denodo,DC=loc
name	DirectoryString	1	Brandon Stevenson
nTSecurityDescriptor	NTSecurityDescriptor	1	D:AI(OA;;RP;4c164200-20c0-11d0-a768-00aa006e0529;;S-1
objectCategory	DN	1	CN=Person,CN=Schema,CN=Configuration,DC=denodo,DC=
objectClass	OID	4	top;person;organizationalPerson;user
objectGUID	OctetString	1	{A468C6E7-9453-4B58-A449-333AFDFEDF6C}
objectSid	Sid	1	S-1-5-21-263618286-132087667-727798575-1109
primaryGroupID	Integer	1	513
pwdLastSet	Integer8	1	10/2/2014 12:06:04 PM
sAMAccountName	DirectoryString	1	bstevenson
sAMAccountType	Integer	1	805306368
userAccountControl	Integer	1	512
userPrincipalName	DirectoryString	1	bstevenson@denodo.loc
uSNChanged	Integer8	1	0x146E0
uSNCreated	Integer8	1	0x36D6
whenChanged	GeneralizedTime	1	10/22/2014 4:21:33 PM
whenCreated	GeneralizedTime	1	4/2/2013 5:54:33 PM

In Active Directory configurations, the recommended attribute is always `sAMAccountName`, because it is **unique** in the Domain (and it is the same identifier used, for example, to authenticate in Windows), but we can choose any other attribute if we want.

Selecting `sAMAccountName`, users will have to use it when using the VDP Administration Tool. In the example, the user “Brandon Stevenson” must use “`bstevenson`” as login.

3.3.3 User search pattern

The last parameter to configure LDAP users is the “`User search pattern`”. This parameter has to be a valid search query to be executed in the Domain Controller to get the **full list of users**. Why is it needed? At login time, when a user tries to connect to Virtual DataPort, the server will do the following:

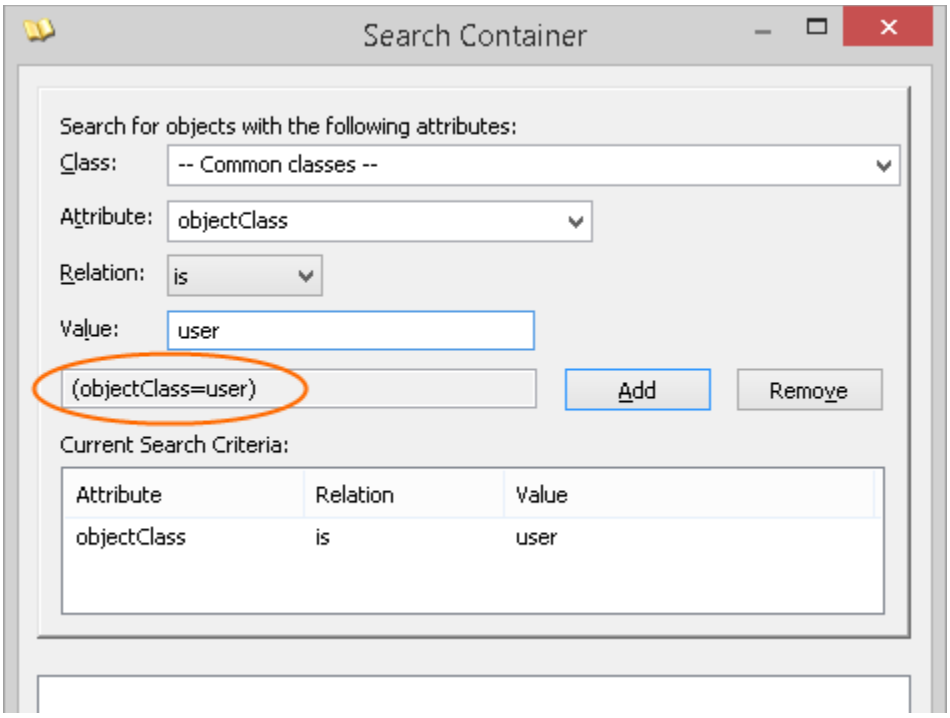
- It connects to the LDAP server using the values (URL path, ...) of the LDAP data source.

- When connected, it searches for the users to get the one having as value of the `sAMAccountName` (or any other attribute configured in the previous section) the same value as the user entered in the Login field when connecting to DataPort.
- If the user is found, the server will get the DistinguishedName (DN) of the user.
- Finally, it delegates to the LDAP server the authentication using the DN and the password provided by the user (if LDAP authentication is ok, then the user is authorized to connect to DataPort).

We are going to use ADexplorer to help us to create the LDAP query (click on `Search > Search Container`). For searching users we have to find an attribute with that information. Returning to the user's attributes screenshot, you can see an attribute with this value:

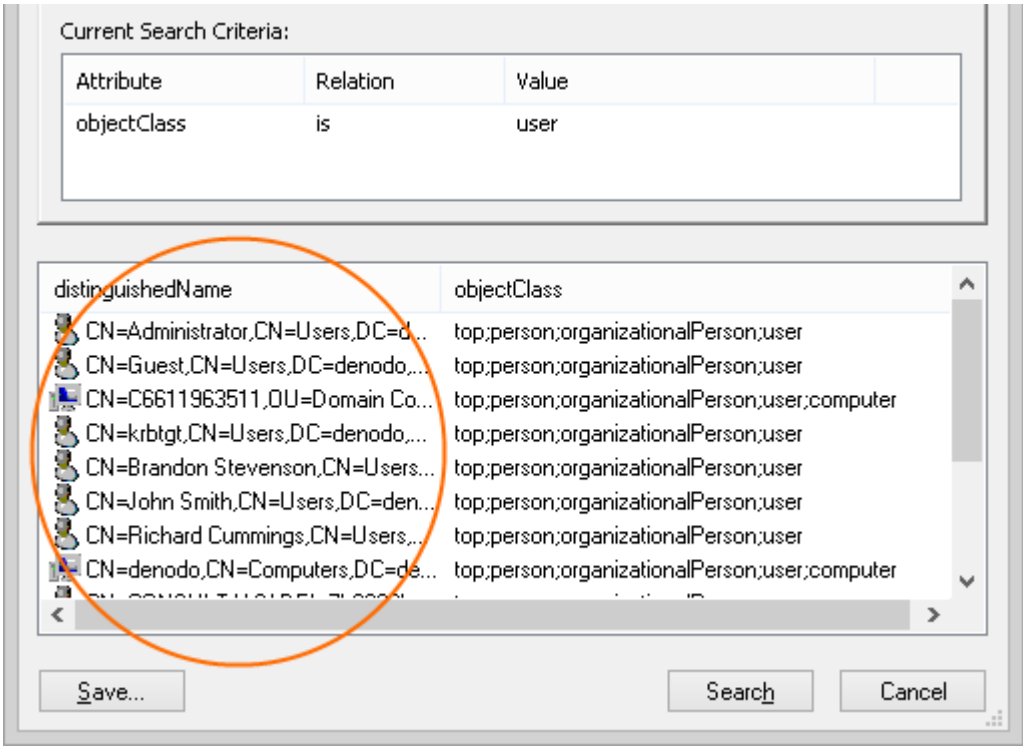
```
objectClass = top;person;organizationalPerson;user
```

It's quite usual to use this attribute to search for users (or computers, or whatever we want to filter). For example, let's try with `objectClass = user` in ADexplorer:

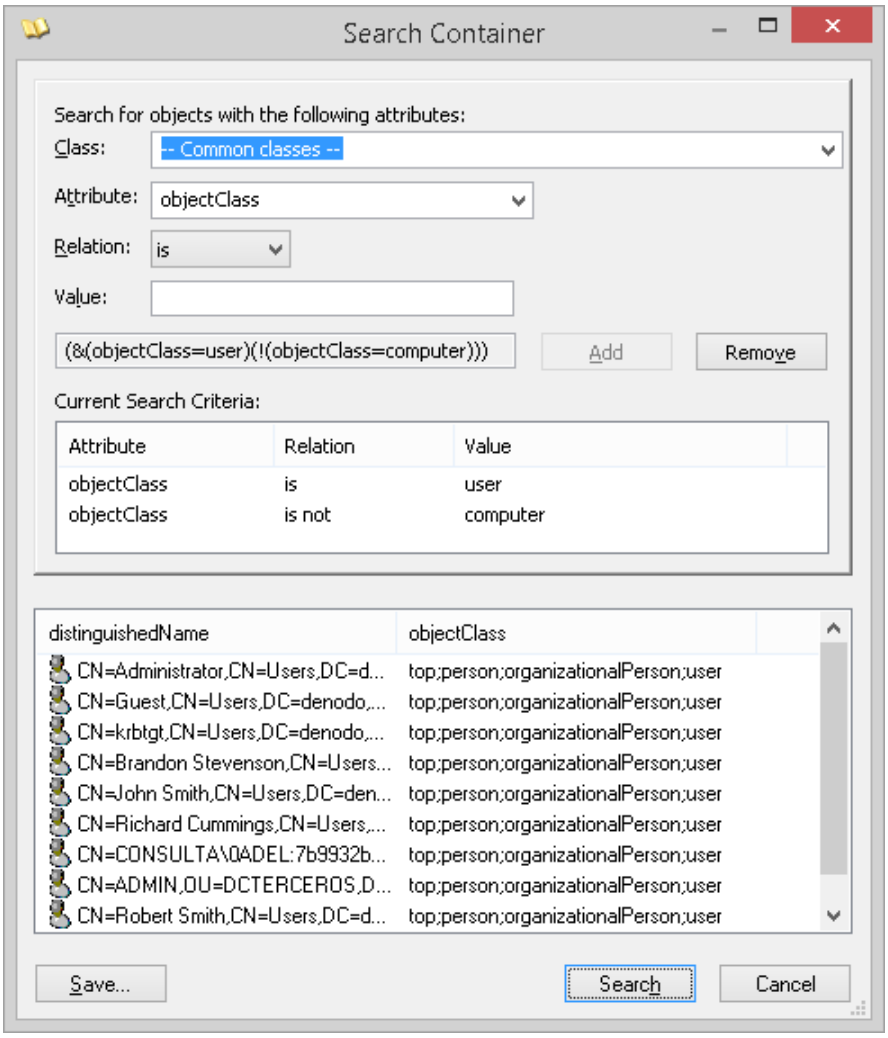


Clicking on the "Add" button, we can see a query (the selected text) which is the LDAP query expression to use in the "User search pattern" parameter in Virtual DataPort.

Clicking on the "Search" button we can test this query:



In this current search, Computers are returned too (they have the same objectClass=user value). In this case we can modify our query to filter the computers out by adding the following to the search: `objectClass is not computer`.



Our final LDAP search query is:

```
(&(objectClass=user)(!(objectClass=computer)))
```

3.3.4 Role base

In the first image of this section we saw that inside `CN=Users` the users were listed among other elements:

- `CN=Allowed RODC Password Replication Group`
- `CN=Cert Publishers`
- `CN=director`
- `CN=employee`
- `CN=Enterprise Admins`
- …

These elements are **groups**. In Active Directory users are members of one or several groups so typically they are used as roles (because people belonging to the same group should have the same privileges over the applications, in our case Virtual DataPort).

As all our groups/roles are in the same container, our **Role base** parameter will be only one node:`CN=Users,DC=denodo,DC=loc`

3.3.5 Attribute with role name

You have to do the same exercise we did with the 'attribute with user name' parameter, but in this case with the role name.

The most commonly used attribute can be the name of the group (usually the attributes `cn` or `name` or `sAMAccountName`). Any of them will work in our case.

Note: the value of this attribute has to be the same as the name of a role created in Virtual DataPort.

3.3.6 Role search pattern

First, we have to remember that we have available the variable `@{USERDN}` (the complete user Distinguished Name) in role searches. As we are using the groups as roles, it is as easy as a search for groups to have the user as member:

```
(&(member=@{USERDN})(objectClass=group))
```

For example if the user `jsmith` tries to connect the Distinguished Name for this user will be:

```
CN=jsmith,CN=Users,DC=denodo,DC=loc
```

and the final query to execute will be:

```
(&(member=CN=jsmith,CN=Users,DC=denodo,DC=loc)(objectClass=group))
```

In short, our parameters are:

- **User base**: `CN=Users,DC=denodo,DC=loc`
- **Attribute with user name**: `sAMAccountName`
- **User search pattern**: `(&(objectClass=user)(!(objectClass=computer)))`
- **Role base**: `CN=Users,DC=denodo,DC=loc`
- **Attribute with role name**: `sAMAccountName`
- **Role search pattern:** `(&(member=@{USERDN})(objectClass=group))`

3.4 IMPORT ROLES FROM THE LDAP SERVER

When a user is authenticated successfully in the LDAP server, Virtual DataPort gets the groups of that user. These groups will be compared with the roles created in VDP. In our example, the user "`John Smith (jsmith)`" is a member of the "`director`" group, so we need to create the role "`director`" in VDP.

This can be done manually but to import a big number of roles from an LDAP server it is recommended to use the **Import Roles tool**. (see section "Creating Roles" of the Virtual DataPort Administration guide). Note that the 'Role Search Pattern' used to search the roles should be indicative of the distinguishing factor in terms of objectClass. For example, using (objectClass=group) allows you to search for the roles in all the groups. You could also choose to exclude a group from the search for roles.

Note: if the role `serveradmin` is imported from the LDAP server it will be ignored for security reasons as this role is the predefined role in VDP for global administrators. To assign global administrator privileges to an user it will be necessary to assign to some of the other roles returned by the LDAP server the `serveradmin` predefined role.

To import just a subset of all the existing roles in an LDAP server it is recommended to manually create the roles instead of iterating on the list of roles. To do that you can use the VDP Admin tool to add the roles one by one or a VQL script to create the roles using the VQL Shell. To create a new role using VQL the following statement can be executed:

```
CREATE ROLE <role_name> '<role_description>';
```

3.5 ASSIGN THE PRIVILEGES TO THE LDAP ROLES TO CONNECT TO THE LDAP DATABASE

The last step to finish the LDAP configuration is to assign privileges to the created roles. It is recommended to use several VDP admin tool sessions simultaneously to simplify the process:

A. VDP Admin tool connected with an admin user that configures the privileges for the role.
B. VDP Admin tool connected with a normal user that belongs to the role whose privileges are being assigned to check if it is working as expected. Any time the privileges of this normal user or its roles are changed the user will have to reconnect to the VDP admin tool so the changes in the configuration take effect.

4 DEBUGGING THE LDAP CONFIGURATION

If you find errors when testing the LDAP configuration, the recommendation is to check the `vdp.log` file under the `<DENODO_HOME>/logs/vdp` folder. If there is not enough information about the error it is possible to set up the following log categories to **trace** using the **logcontroller** stored procedure. For example, the following commands:

```
call logcontroller(
      'com.denodo.vdb.security.LDAPDatabaseAuthenticator',
      'trace');

call logcontroller(
       'com.denodo.vdb.catalog.user.User',
      'trace');
```

will enable trace information in the logs for the categories relevant to the LDAP authentication, and:

```
call logcontroller(
      'com.denodo.vdb.security.DefaultLDAPUserGroupRetriever',
      'trace');

call logcontroller(
      'com.denodo.vdb.engine.wrapper.raw.ldap.executor.LDAPExecutor',
      'trace');

call logcontroller(
      'com.denodo.vdb.catalog.role.Role',
      'trace');
```

will enable trace information in the logs for the categories relevant to the roles.

It is important to remember to change these categories back to error once the error has been diagnosed and solved.

For example, let's see the log written when the user "`John Smith (jsmith)`" tries to login:

```
52270 [RMI TCP Connection(8585)-10.0.20.154] TRACE 20141021125041055
com.denodo.vdb.engine.wrapper.raw.ldap.executor.LDAPExecutor  - Executing
LDAPrequest. [base search = CN=Users,DC=denodo,DC=loc, filter =
(&(&(objectClass=user)(!(objectClass=computer)))(sAMAccountName=jsmith)),
paging = false, page size = 1000, cookie = null]
```

- The server makes a query to the ldap data source searching for objects having `user` but not `computer` as objectClass and having `jsmith` as `sAMAccountName`.

```
353952426 [RMI TCP Connection(8585)-10.0.20.154] DEBUG 20141021125041211
com.denodo.vdb.engine.wrapper.raw.ldap.executor.LDAPExecutor  - Pages
readed: 0, with pagesize: 1000 and retrieved elements with last page: 1
```

- Number of retrieved elements of the LDAP server (in this case only one)

```
353952426 [RMI TCP Connection(8585)-10.0.20.154] DEBUG 20141021125041211
com.denodo.vdb.security.LDAPDatabaseAuthenticator  - User found: CN=John
Smith,CN=Users,DC=denodo,DC=loc
```

- Distinguished Name (DN) of the retrieved element (`CN=John Smith,CN=Users,DC=denodo,DC=loc`)

```
353952703 [RMI TCP Connection(8585)-10.0.20.154] DEBUG 20141021125041488
com.denodo.vdb.security.LDAPDatabaseAuthenticator  - Authentication
successfully
```

- Password is checked against LDAP server using the user DN (in this example, success)

```
353952703 [RMI TCP Connection(8585)-10.0.20.154] DEBUG 20141021125041488
com.denodo.vdb.security.DefaultLDAPUserGroupRetriever  - Searching LDAP
groups with User: CN=John Smith,CN=Users,DC=denodo,DC=loc Base:
CN=Users,DC=denodo,DC=loc Filter: (distinguishedname=CN=John
Smith,CN=Users,DC=denodo,DC=loc)

353952703 [RMI TCP Connection(8585)-10.0.20.154] TRACE 20141021125041488
com.denodo.vdb.engine.wrapper.raw.ldap.executor.LDAPExecutor  - Executing
LDAPrequest. [base search = CN=Users,DC=denodo,DC=loc, filter =
(distinguishedname=CN=John Smith,CN=Users,DC=denodo,DC=loc), paging =
false, page size = 1000, cookie = null]
```

- Role search using the DN.

```
353952835 [RMI TCP Connection(8585)-10.0.20.154] DEBUG 20141021125041620
com.denodo.vdb.security.DefaultLDAPUserGroupRetriever  - Group attribute
name: sAMAccountName: director

353952850 [RMI TCP Connection(8585)-10.0.20.154] DEBUG 20141021125041635
com.denodo.vdb.security.DefaultLDAPUserGroupRetriever  - Processing group
director

353952850 [RMI TCP Connection(8585)-10.0.20.154] DEBUG 20141021125041635
com.denodo.vdb.engine.wrapper.raw.ldap.executor.LDAPExecutor  - Pages
readed: 0, with pagesize: 1000 and retrieved elements with last page: 1
```

- Processed groups/roles (in this example, one group ‘director’ was found)

Following the log you can see where the error is coming from:

- User not found in the LDAP server
- Incorrect password
- The groups are not retrieved correctly.
- Groups are retrieved but the role is not created in Virtual DataPort.
- ...

5 ADVANCED FEATURES

5.1 WORKING WITH HIERARCHICAL ELEMENTS

In some environments, you may need to define some LDAP groups which do not have the users as members as roles in VDP. That configuration is hierarchical: users belong to some groups and those groups are members of other groups.

In this specific scenario, a special syntax is needed in the search queries in order to retrieve all user groups recursively. For example:

Role search pattern: `member:1.2.840.113556.1.4.1941:=@USERDN`

5.2 LDAP ELEMENTS WITH UNICODE CHARACTERS

If your LDAP elements (user / roles) have unicode characters, you will need to activate in Virtual DataPort "**Unicode**" as identifiers charset at global server level (`Administration > Server configuration > Identifiers charset`).

Note: if your preference is to use the Restricted charset in the Administration Tool when creating views, you can always restore the Identifiers charset of an individual database to "`Restricted`" (`Administration > Database management > <database> > Edit`)

5.3 ASSIGN PRIVILEGES TO SINGLE USERS

If you are in a situation where you need to use permissions on a user level then you can edit the LDAP query to search for roles.

In our previous example, the user "`jsmith`" has the role "`director`" in VDP. Editing the role search pattern and use something similar to:

```
(|(distinguishedName=@{USERDN})(member=@{USERDN}))
```

The search query will return nodes containing `"CN=John Smith, CN=Users,DC=denodo,DC=loc"` as member **OR** nodes having as distinguishedName `"CN=John Smith, CN=Users,DC=denodo,DC=loc"` (the "`John Smith`" user itself). So in this case, he will have "*director*" and "*jsmith*" as roles in VDP (you will have to create this new specific role for this specific user).

References
LDAP Authentication at server level
Importing LDAP roles in Virtual DataPort