



# LDAP BASICS AND ITS CONFIGURATION

LDAP COMMANDS-Linux and Windows

## Abstract

This document describes the basics of LDAP and how to install and configure it in Linux and windows

Sahu, Haramohan  
Hara.sahu@gmail.com

## Contents

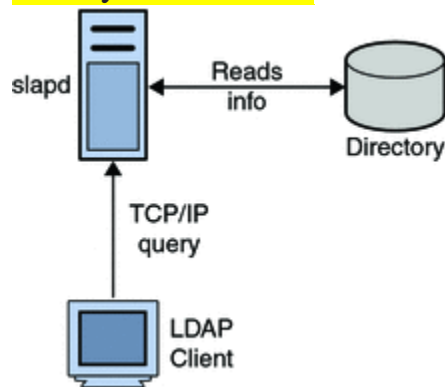
What is LDAP? .....	2
What is a Directory Service? .....	4
Directory Servers .....	4
Why Do We Want Directory Services? .....	5
LDAP vs. Database .....	5
Directory Information Tree Hierarchical structure Directory Information Tree (DIT) .....	10
How is information stored? .....	13
Connecting and searching LDAP servers.....	15
FAQ.....	16
What is bind DN and base DN? .....	16
What is Bind DN?.....	16
<b>Base DN Details for LDAP → command in windows</b> .....	17
<b>Administrator Bind DN Details for LDAP</b> .....	17
Finding the User Base DN .....	19
Finding the Group Base DN .....	19
All DSQUERY Commands.....	20
Man of dsquery .....	21
Search Query → in Linux .....	22
How ldap search? .....	25
LDAP Referrals:.....	26
Steps to Set Ldap in RHEL .....	28
Steps to setup ldap in Windows.....	33
How to configure DFM:.....	50
DFM community links: .....	51

## What is LDAP?

LDAP, or lightweight directory access protocol, is a communications protocol that defines the methods in which a directory service can be accessed. More broadly speaking, LDAP shapes the way that the data within a directory service should be represented to users, defines requirements for the components used to create data entries within a directory service, and outlines the way that different primitive elements are used to compose entries.

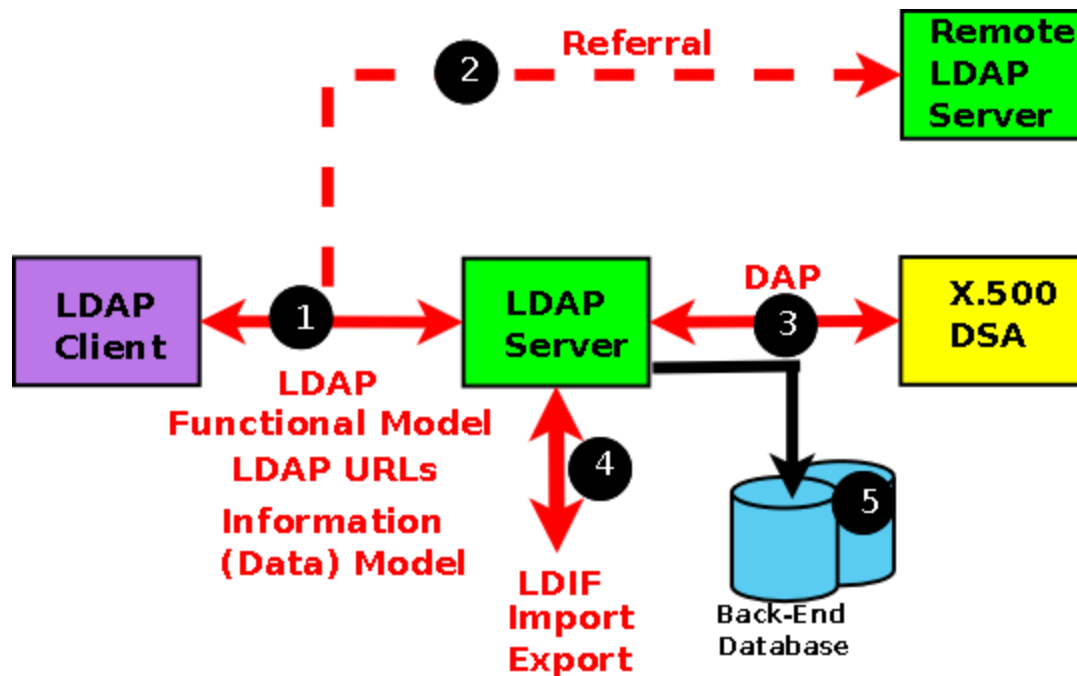
Since LDAP is an open protocol, there are many different implementations available. The OpenLDAP project is one of the most well supported open source variants.

LDAP is a protocol used to communicate with a directory database to query, add or modify information.



LDAP allows clients to access different directory services based on entries. These LDAP entries are available to users and other applications based on access controls.

The scope of the LDAP standards is shown in the diagram below. The red stuff (1,2, 3, 4) is defined in the protocol (the various RFCs that define LDAP). What happens inside the black boxes (or in this case the green, yellow and mauve boxes) and on the black line to the Database(s) (5) is 'automagical' and outside the scope of the standards.



Each component is briefly described here, in a bit more detail below and in excruciating detail in subsequent chapters. But there are four important points first:

1. LDAP does not define how data is stored, only how it is accessed. BUT most LDAP implementations do use a standard database as a back-end and indeed OpenLDAP offers a choice of back-end database support.
2. When you talk to an LDAP server you have no idea where the data comes from: in fact the whole point about the standard is to hide this level of detail. In theory the data may come from one OR MORE local databases or one OR MORE X.500 services (though these are about as rare as hen's teeth these days). Where and how you access the data is an implementation detail and is only important when you define the operational configuration of your LDAP server(s).
3. Keep the two concepts - access to the LDAP service and operation of the LDAP service - very clearly separate in your mind. When you design a directory based system figure out what you want it to do (the data organization) and forget the implementation. Then figure out as a second phase where the data is and how and where you want to store it - during LDAP operational configuration.
4. A number of commercial database products provide an LDAP view (an LDAP wrapper or an LDAP abstraction) of relational or other database types.

## What is a Directory Service?

A directory service is used to store, organize, and present data in a key-value type format. Typically, directories are optimized for lookups, searches, and read operations over write operations, so they function extremely well for data that is referenced often but changes infrequently.

The data stored in a directory service is often descriptive in nature and used to define the qualities of an entity. An example of a physical object that would be well represented in a directory service is an address book.

Each person could be represented by an entry in the directory, with key-value pairs describing their contact information, place of business, etc. Directory services are useful in many scenarios where you want to make qualitative, descriptive information accessible.

### Directory Service Limitations

- Object-oriented, no tables, no joins
- Standard data model limitations
- Management tools are ... cumbersome
- LDAP is a database, not authentication server
- Single directory myth
- You will need more components
  - Directory server(s)
  - Access management (SSO)
  - Identity management

## Directory Servers

Since LDAP is an open standard protocol, all of the information needed to create an LDAPv3-compliant server is freely available.

### **Traditional LDAP Directory Servers:**

LDAP directory servers that you can run yourself, on your own equipment or in the cloud:

1. 389 Directory Server (formerly Fedora Directory Server)
2. ApacheDS
3. CA Directory (formerly CA eTrust Directory)
4. ForgeRock Directory Services
5. Fusion Directory (tailored for educational deployments)

6. GLAuth
7. IBM Security Directory Server (formerly IBM Tivoli Directory Server and IBM SecureWay Directory)
8. Isode M-Vault LDAP/X.500 Server
9. Microsoft Active Directory
10. NetIQ eDirectory (formerly Novell eDirectory)
11. OpenDJ
12. OpenLDAP
13. Oracle Internet Directory
14. Oracle Unified Directory
15. Ping Identity Directory Server (formerly UnboundID Directory Server)
16. Red Hat Directory Server
17. Symas OpenLDAP (an enhanced version of OpenLDAP with available commercial support)

Native LDAP Server Storing data in LMDB databases (or other backends) Tailor-made database and indexing Excellent performance Access protocols: LDAP  
Written in C, long source code history

### Why Do We Want Directory Services?

- They are fast! Really fast. When reading. Faster than the fastest relational databases But slow when writing (approx 10 times)
- Low resource consumption Approx 10 times lower than relational DBs
- Scaling ad nauseam 1M entries is nothing. 1B is still easy.
- Easy to replicate the data High availability, performance, scaling

### LDAP vs. Database

LDAP is characterized as a write-once-read-many-times service. That is to say, the type of data that would normally be stored in an LDAP service would not be expected to change on every access. To illustrate: LDAP would not be suitable for maintaining banking transaction records since, by their nature, they change on almost every access (transaction). LDAP would, however, be eminently suitable for maintaining details of the bank branches, hours of opening, employees, and so on which change far less frequently.

### Basic LDAP Data Components

LDAP is a protocol used to communicate with a directory database to query, add or modify information. However, this simple definition misrepresents the complexity of the systems that support this protocol. The way that LDAP displays data to users is very dependent upon the interaction of and relationship between some defined structural components.

The following is a list of LDAP-specific terms that are used in ldap context.

- **Entry:** An LDAP entry is a collection of information about an entity which is a single unit within an LDAP directory. Each entry is identified by its unique Distinguished Name (DN). Each entry consists of three primary components: a distinguished name, a collection of attributes, and a collection of object classes. Each of these is described in more detail below.

1. **DNs and RDNs:**
2. **Attributes:**
3. **Object Classes:**

**DNs and RDNs:** An entry's distinguished name, often referred to as a DN, uniquely identifies that entry and its position in the directory information tree (DIT) hierarchy. The DN of an LDAP entry is much like the path to a file on a filesystem. A distinguished name (usually just shortened to "DN") uniquely identifies an entry and describes its position in the DIT. A DN is much like an absolute path on a filesystem, except whereas filesystem paths usually start with the root of the filesystem and descend the tree from left to right, LDAP DNs ascend the tree from left to right. For example, the DN "uid=john.doe,ou=People,dc=example,dc=com" represents an entry that is immediately subordinate to "ou=People,dc=example,dc=com" which is itself immediately subordinate to the entry "dc=example,dc=com".

DNs are comprised of zero or more comma-separated components called relative distinguished names, or RDNs. For example, the DN "uid=john.doe,ou=People,dc=example,dc=com" has four RDNs:

- uid=john.doe
- ou=People
- dc=example
- dc=com

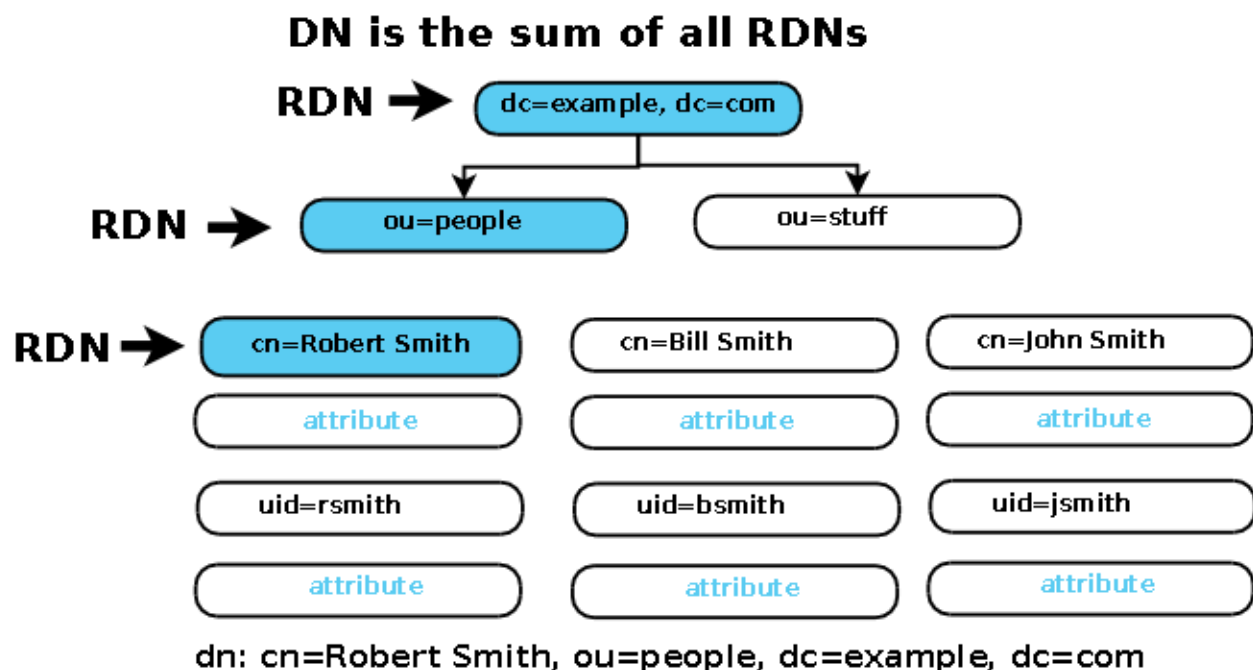
There may be multiple ways of representing some or all of the values of the RDN components (e.g., differences in capitalization may be considered insignificant). For example, all of the following are valid ways of representing the same DN:

1. dc=example,dc=com
2. dc=example, dc=com
3. dc = example , dc = com
4. DC=EXAMPLE,DC=COM
5. 0.9.2342.19200300.100.1.25=Example,0.9.2342.19200300.100.1.25=Com

The DN is written LEFT to RIGHT.

The DN is composed of a series of RDNs (Relative Distinguished Names) which are the unique (or unique'ish) attributes at each level in the DIT hierarchy.

The following diagram illustrates building up the DN from the RDN's:

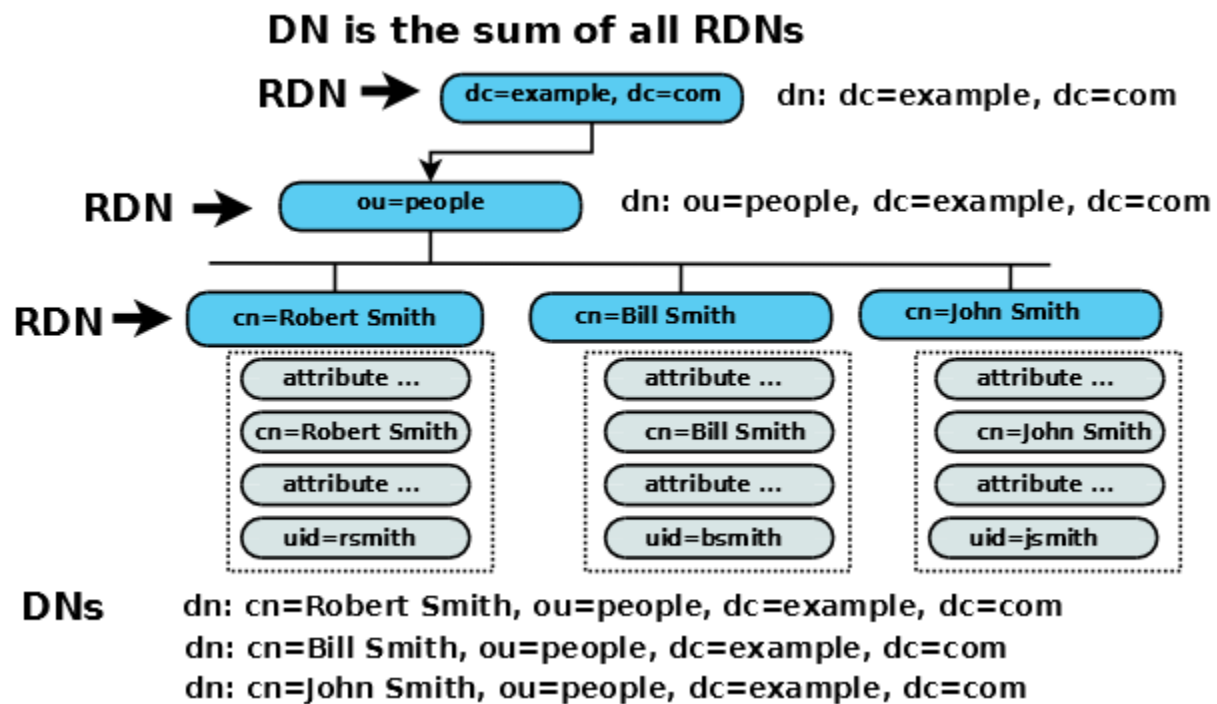


In the above example we have selected the **cn (commonName)** attribute as our RDN because it is unique at this level in the directory. This gives a full DN of:

**DN: cn=Robert Smith,ou=people,dc=example,dc=com**

**Some more example:**





## Attributes:

Information directly associated with an entry. For example, if an organization is represented as an LDAP entry, attributes associated with this organization might include an address, a fax number, and so on. Similarly, people can be represented as entries with common attributes such as personal telephone number or email address.

Each attribute has a name and one or more values. The names of the attributes are mnemonic strings, such as cn for common name, or mail for email address.

For example, a company may have an employee directory. Each entry in the employee directory represents an employee. The employee entry contains such information as the name, email address, and phone number, as shown in the following example:

<b>cn: John Doe</b>
<b>mail: johndoe@sun.com</b>
<b>mail: jdoe@stc.com</b>

telephoneNumber: 471-6000 x.1234

## Attribute Abbreviation

- User id : uid
- Common Name ; cn
- Surname : sn
- Location : l
- Organizational Unit : ou
- Organization : o
- Domain Component : dc
- State : st
- Country : c
- Street address : street

The screenshot shows an LDAP browser interface. On the left is a directory tree with the following structure:

- dc=viyademo,dc=com (2)
  - ou=gelcorp (2)
    - ou=groups (5)
      - cn=gelcorp
      - cn=gelcorpadmins
      - cn=gelcorpshr
      - cn=gelcorpmanagers
      - cn=gelcorpsales
    - ou=users (24)
      - uid=Ahmed
      - uid=Amanda
      - uid=Delilah
      - uid=Douglas
      - uid=Hamish
      - uid=Hazel

The 'uid=Ahmed' entry is selected. On the right, the 'DN: uid=Ahmed,ou=users,ou=gelcorp,dc=viyademo,dc=com' entry is displayed with the following attributes:

Attribute Description	Value
objectClass	inetOrgPerson (structural)
objectClass	organizationalPerson (structural)
objectClass	top (abstract)
cn	Ahmed
sn	Ahmed
displayName	Ahmed
employeeNumber	P219
l	Cary
mail	Ahmed@gelcorp.com
o	GEL Corp
title	Administrator
uid	Ahmed
userPassword	SSHA hashed password

Two green boxes with arrows point to the interface:

- A box labeled 'Entries' points to the 'uid=Ahmed' entry in the directory tree.
- A box labeled 'Attribute names and values' points to the attribute list on the right.

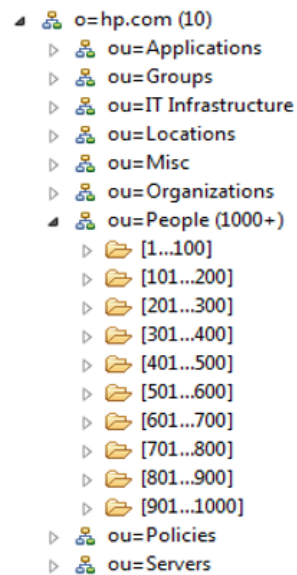
## Object Classes:

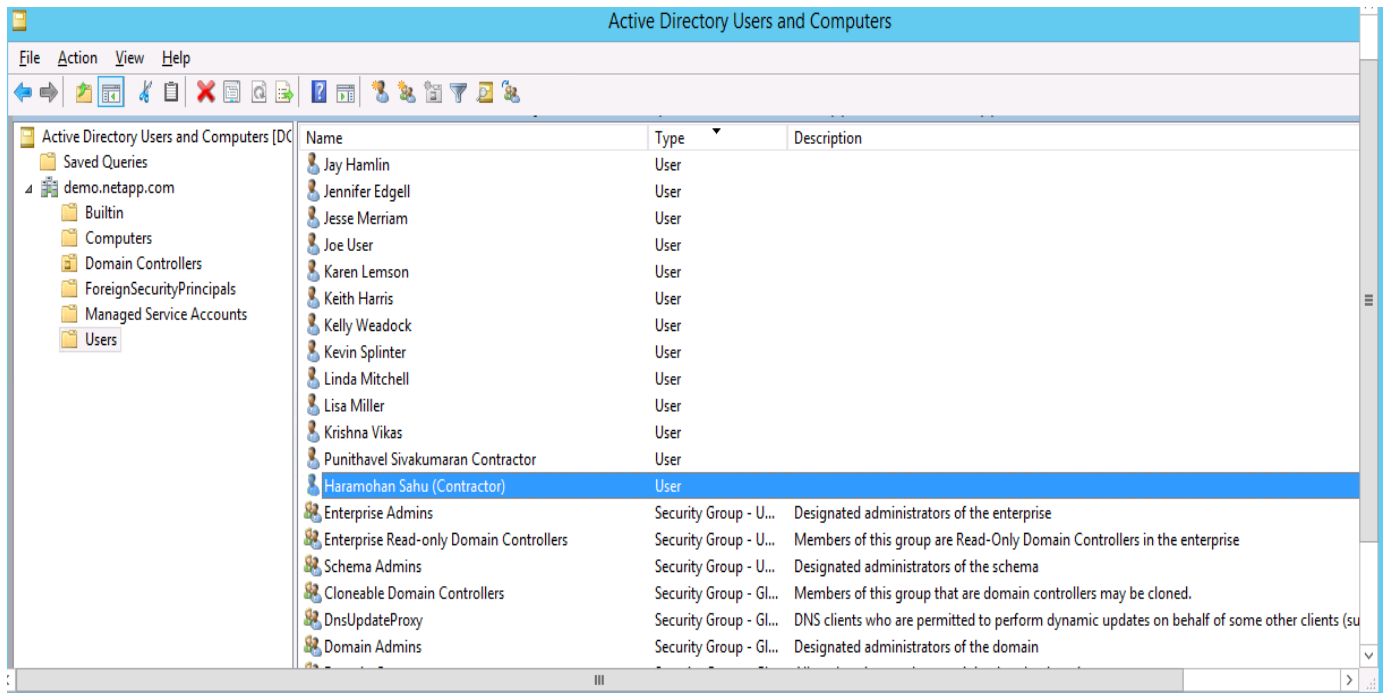
This is a special type of attribute. All objects in LDAP must have an objectClass attribute. The objectClass definition specifies which attributes are required for each LDAP object, and it specifies the object classes of an entry. The

values of this attribute may be modified by clients, but the objectClass attribute itself cannot be removed.

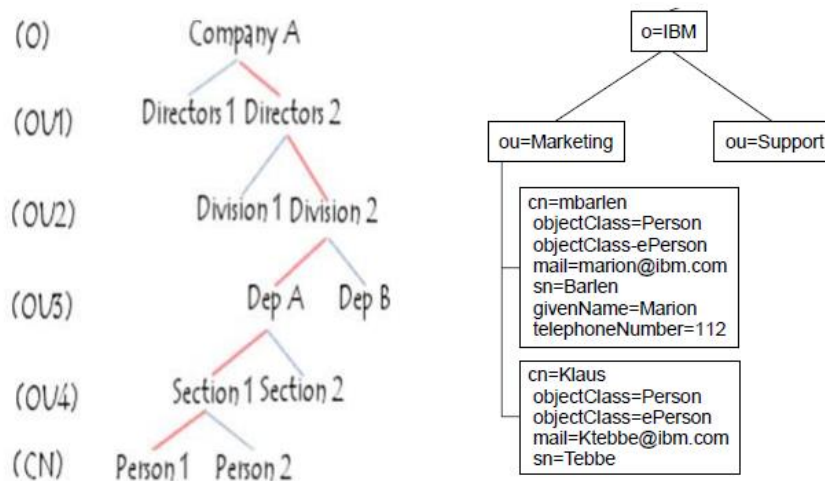
## Directory Information Tree Hierarchical structure Directory Information Tree (DIT)

### HP Directory Information Tree (DIT).

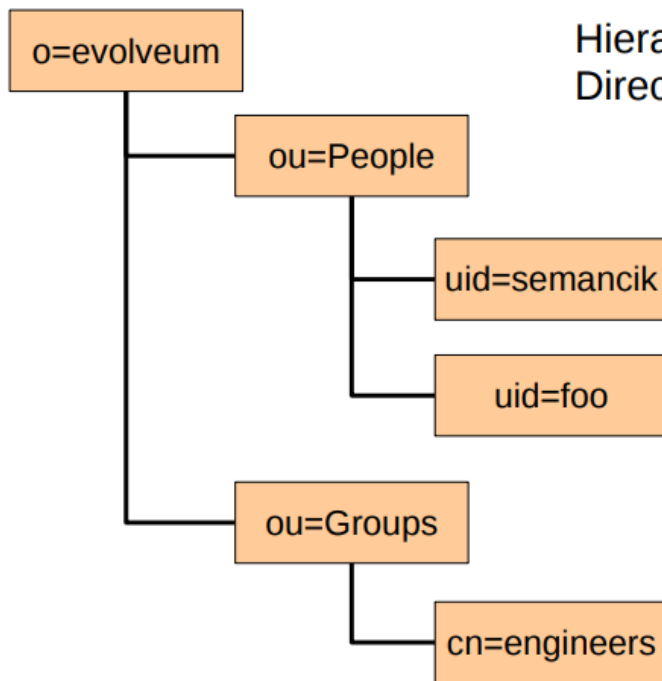




- Presents information in the form of a hierarchical **tree** structure called a **DIT** (Directory Information Tree).

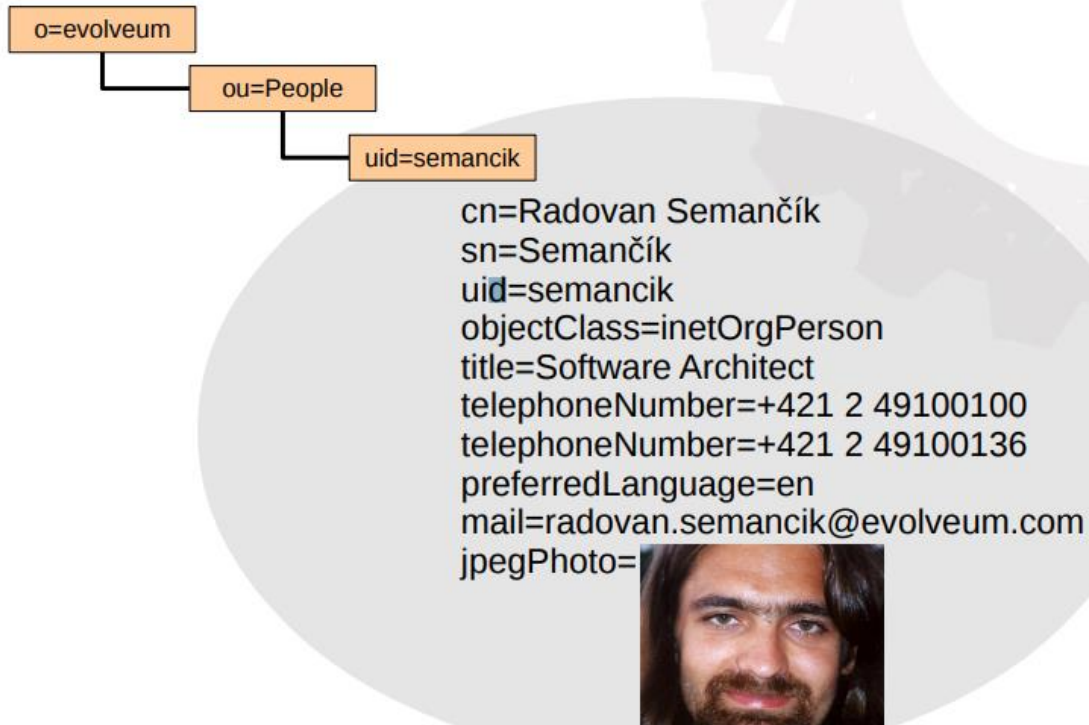


# Directory Information Tree



Hierarchical structure  
Directory Information Tree (DIT)

# Objects & Attributes



How is information stored?

- LDAP is a **hierachical (tree-based)** database.
- Information is stored as **key-value pairs**.
- The tree structure is basically **free-form**. Every organisation can choose how to arrange the tree for themselves, although there are some commonly used patterns.

## The tree

An example of an LDAP tree structure (*some otherwise required attributes are left out for clarity!*):

```
dc=com
  dc=megacorp
    ou=people
      uid=jjohnson
        objectClass=inetOrgPerson,posixAccount
        cn=John Johnson
```

```
uid=jjohnson
mail=j.johnson@megacorp.com
uid=ppeterson
objectClass=inetOrgPerson,posixAccount
cn=Peter Peterson
uid=ppeterson
mail=p.peterson@megacorp.com
```

- Each leaf in the tree has a specific unique path called the **Distinguished Name (DN)**. For example: `uid=ppeterson,ou=people,dc=megacorp,dc=com`
- Unlike file paths and most other tree-based paths which have their roots on the left, **the Distinguished Name has the root of the tree on the right**.
- Instead of the conventional path separators such as the dot ( . ) or forward-slash ( / ), **the DN uses the comma ( , ) to separate path elements**.
- Unlike conventional paths (e.g. `/com/megacorp/people/ppeterson`), **the DN path includes an attribute type for each element in the path**. For instance: `dc=`, `ou=` and `uid=`. These are abbreviations that specify the type of the attribute. More on attribute types in the Entry chapter.
- It is common to arrange the tree in a globally unique way, using `dc=com,dc=megacorp` to specify the organisation.
- **Entries are parts of the tree that actually store information**. In this case: `uid=jjohnson` and `uid=ppeterson`.

## Entries

An example entry for DN `uid=jjohnson,ou=people,dc=megacorp,dc=com` (*some otherwise required attributes are left out for clarity!*):

```
objectClass=inetOrgPerson,posixAccount
cn=John Johnson
uid=jjohnson
mail=j.johnson@megacorp.com
```

- An entry has an Relative Distinguished Name (RDN). **The RDN is a unique identifier for the entry in that part of the tree**. For the entry with Distinguished Name (DN) `uid=jjohnson,ou=people,dc=megacorp,dc=com`, the RDN is `uid=jjohnson`.
- **An entry stores key/value pairs**. In LDAP lingo, these are called **attribute types** and **attribute values**. Attribute types are sometimes abbreviations. In this case, the attribute types are **cn=** (CommonName), **uid=** (UserID) and **mail=**.

- **Keys may appear multiple times**, in which case they are considered as a list of values.
- **An entry has one or more objectClasses.**
- **Object classes are defined by schemas, and they determine which attributes must and may appear in an entry.** For instance, the `posixAccount` object class is defined in the `nis.schema` and must include `cn`, `uid`, etc.
- Different object classes may define the same attribute types.

## Connecting and searching LDAP servers

The most common action to perform on LDAP servers is to search for information in the directory. For instance, you may want to search for a username to verify if they entered their password correctly, or you may want to search for Common Names (CNs) to auto-complete names and email addresses in your email client. In order to search an LDAP server, we must perform the following:

1. Connect to the LDAP server
2. Authenticate against the LDAP server so we are allowed to search. This is called **binding**. Basically it's just logging in. We bind against an LDAP server by specifying a **user's DN and password**. This can be confusing because there can be DN's/password with which you can bind in the LDAP, but also user/passwords which are merely stored so that other systems can authenticate users using the LDAP server.
3. Specify which sub-part of the tree we wish to search. This is called **the Base DN (Base Distinguished Name)**. For example: `ou=people,dc=megacorp,dc=com`, so search only people. Different bind DN's may search different parts of the tree.
4. Specify how deep we want to search in the tree. This is called **the level**. The level can be: `BaseObject` (search just the named entry, typically used to read one entry), `singleLevel` (entries immediately below the base DN), `orWholeSubtree` (the entire subtree starting at the base DN).
5. Specify what kind of entries we'd like to search for. This is called **the filter**. For example, `(objectClass=*)` will search for ANY kind of object



`class. (objectClass=posixAccount)` will only search for entries of the `posixAccount` object class.

Here's an example of connecting, binding and searching an LDAP server using the `ldapsearch` commandline util:

```
$ ldapsearch -W -h ldap.megacorp.com -D
"uid=ldapreader,dc=megacorp,dc=com"
  -b ou=people,dc=megacorp,dc=com "(objectclass=*)"
password: *****
```

- `-W` tells `ldapsearch` to prompt for a password.
- `-h` is the hostname of the LDAP server to connect to.
- `-D` is the Distinguished Name (DN), a.k.a the username, with which to connect. In this case, a special `ldapreader` account.
- `-b` is the Base DN, a.k.a the subtree, we want to search.

Finally, we specify a search filter: `"(objectclass=*)"`. This means we want to search for all object classes.

The previous example, but this time in the Python programming language:

```
import ldap
l = ldap.initialize('ldap://ldap.megacorp.com:389')

l.bind('uid=ldapreader,dc=megacorp,dc=com', 'Myp4ssw0rD')
l.search_s('ou=people,dc=megacorp,dc=com', ldap.SCOPE_SUBTREE,
          filterstr="(objectclass=*)" )
```

**BaseDN** the entry in the tree from which the LDAP server starts its search.

## FAQ

### What is bind DN and base DN?

An Active Directory Administrator **Bind DN & Base DN** is needed to use our LDAP Authentication and/or Import Users. The Admin **Bind DN** allows the LDAP connection to gain access into the Active Directory while the **Base DN** tells it where to look for the requested information.

### What is Bind DN?

**The Bind DN** is comprised of **the** user and **the** location of **the** user in **the** LDAP directory tree. ... Therefore, **the Bind DN** is:  
CN=user1,CN=Users,DC=example,DC=com. If **the** domain was

example.net, **the** syntax would be DC=example,DC=net. DC is used for **the** domain portion, and CN is used for **the** User credentials

## Base DN Details for LDAP → command in windows

The Base DN is the starting point an LDAP server uses when searching for users authentication within your Directory.

Example: DC=example-domain,DC=com

1. In the Start menu, search for "cmd"
2. Right click on Command Prompt and select Run as Administrator
3. The servers Command Prompt will open, in the prompt run dsquery \*

```
C:\Users\Administrator>dsquery *
```

4. The first output displayed is your Base DN:

```
"DC=example-domain,DC=com"
```

5. Take note of your Base DN, it will be needed for later steps.

## Administrator Bind DN Details for LDAP

A domain administrator details, including Bind DN and password, is needed for our Control Panel to communicate with the LDAP server.

Example: CN=example-user,CN=Users,DC=example-domain,DC=com.

To find the Bind DN for the administrative user and/or any user:

1. In the Start menu, search for cmd or Command Prompt
2. Right click on Command Prompt and select Run as Administrator
3. The servers Command Prompt will open, in the prompt run dsquery user -name <Users Name> \*
  - The wild card \* is needed for users with spaces in their name, ie. John Smith.

```
C:\Users\Administrator>dsquery user -name <Users Name> *
```

4. Take note of the syntax of the Bind DN account for the domain administrator. It should resemble the following:

```
CN=Users-Name,CN=Users,DC=example-domain,DC=com *
```

- The Base DN is where the PAN will start searching in the directory structure.
- The Bind DN is the username that will be used to do the searching and request the authentication.

**Note:** In Active Directory, a blank folder icon represent Containers (CN) while folders with icons are Organizational Units (OU).



```
C:\Users\Administrator>dsquery user dc=demo,dc=netapp,dc=com -name administrator
"CN=Administrator,CN=Users,DC=demo,DC=netapp,DC=com"
```

```
C:\Users\Administrator>
```

```
C:\Users\Administrator>dsquery user -name administrator
"CN=Administrator,CN=Users,DC=demo,DC=netapp,DC=com"
```

```
C:\Users\Administrator>
```

```
C:\Users\Administrator>dsquery user CN=Users,DC=demo,DC=netapp,DC=com -name h*
```

```
"CN=Haramohan Sahu (Contractor),CN=Users,DC=demo,DC=netapp,DC=com"
```

```
"CN=Hemant Mahawar,CN=Users,DC=demo,DC=netapp,DC=com"
```

```
"CN=Henriette Andersen,CN=Users,DC=demo,DC=netapp,DC=com"
```

```
"CN=Henrik Jensen,CN=Users,DC=demo,DC=netapp,DC=com"
```

```
C:\Users\Administrator>
```

## Finding the User Base DN

1. Open a Windows command prompt.
2. Type the command:

```
dsquery user -name <known username>
```

Example: If you are searching for all users named "John", you can enter the username as John\* to get a list of all users who's name is John.

The result will look like:

```
"CN=John.Smith,CN=Users,DC=MyDomain,DC=com"
```

3. - In Symantec Reporter's LDAP/Directory settings, when asked for a User Base DN, enter:

```
CN=Users,DC=MyDomain,DC=com
```

## Finding the Group Base DN

1. Open a Windows command prompt.
2. Type the command:

```
dsquery group -name <known group name>.
```

Example: If you are searching for a group called Users, you can enter the group name as Users\* to get a list of all groups who's name contains "Users"

The result will look like:

```
"CN=Users,CN=Builtin,DC=MyDomain,DC=com"
```

3. In Symantec Reporter's LDAP/Directory settings, when asked for a User Base DN, enter:

```
CN=Users,CN=Builtin,DC=MyDomain,DC=com
```

## More explanation

- Open a Windows command prompt.
- Type the command: `dsquery user -name <known username>`

(Example: If I were searching for all users named John, I could enter the username as John\* to get a list of all users who's name is John)

example: `dsquery user -name *john*`

– The result will look like:

“CN=John.Smith,CN=Users,DC=MyDomain,DC=com”

– If you need this information for configurations like Blue Coat Reporter's LDAP/Directory settings, when asked for a User Base DN, you would enter:

`CN=Users,DC=MyDomain,DC=com`

### **To find the Group Base DN:**

– Open a Windows command prompt

– Type the command: `dsquery group -name <known group name>.`

(Example: If I were searching for a group called Users, I could enter the group name as Users\* to get a list of all groups who's name contains “Users”)

– The result will look like: “CN=Users,CN=Builtin,DC=MyDomain,DC=com”

– In Blue Coat Reporter's LDAP/Directory settings, when asked for a User Base DN, you would enter: CN=Users,CN=Builtin,DC=MyDomain,DC=com.

### All DSQUERY Commands

<https://www.computerperformance.co.uk/logon/dsquery/>

<https://social.technet.microsoft.com/wiki/contents/articles/2195.active-directory-dsquery-commands.aspx>

Description: This tool's commands suite allow you to query the directory according to specified criteria. Each of the following dsquery commands finds objects of a specific object type, with the exception of dsquery \*, which can query for any type of object:

Type "dsquery computer" -  
finds computers in the directory.

Type "dsquery contact" -  
finds contacts in the directory.

Type "dsquery subnet" -  
finds subnets in the directory.

Type "dsquery group" -  
finds groups in the directory.

Type "dsquery ou" -

finds organizational units in the directory.

Type "dsquery site" -

finds sites in the directory.

Type "dsquery server" -

finds AD DCs/LDS instances in the directory.

Type "dsquery user" -

finds users in the directory.

Type "dsquery quota" -

finds quota specifications in the directory.

Type "dsquery partition" -

finds partitions in the directory.

Type "dsquery \*" -

finds any object in the directory by using a generic LDAP query.

[Man of dsquery](#)

See also:

dsquery computer /? - help for finding computers in the directory.

dsquery contact /? - help for finding contacts in the directory.

dsquery subnet /? - help for finding subnets in the directory.

dsquery group /? - help for finding groups in the directory.

dsquery ou /? - help for finding organizational units in the directory.

dsquery site /? - help for finding sites in the directory.

dsquery server /? - help for finding AD DCs/LDS instances in the directory.

dsquery user /? - help for finding users in the directory.

dsquery quota /? - help for finding quotas in the directory.

dsquery partition /? - help for finding partitions in the directory.

dsquery \* /? - help for finding any object in the directory by using a generic LDAP query.

Directory Service command-line tools help:

dsadd /? - help for adding objects.

dsget /? - help for displaying objects.

dsmod /? - help for modifying objects.

dsmove /? - help for moving objects.

dsquery /? - help for finding objects matching search criteria.

dsrm /? - help for deleting objects.

## Search Query → in Linux

The argument for `-D` is the account you use to bind against the LDAP server

- Just like a relation DB, you can perform queries to retrieve the nodes and their attributes. For example, you can try to find all nodes with `cn=Kent Tong` (the 'filter') under the node whose dn is `dc=foo,dc=com` (the 'base' of the search):

```
# ldapsearch -b dc=foo,dc=com -H ldap:// -Y external 'cn=Kent Tong'
```

- You may try to find all nodes whose objectClass is 'person' and whose cn contains the word 'Kent':

```
# ldapsearch -b dc=foo,dc=com ... "(&(objectClass=person)(cn=*Kent*))"
```

- You may tell it to retrieve only a certain attributes:

```
# ldapsearch -b dc=foo,dc=com ... "(&(objectClass=person)(cn=*Kent*))" dn sn
ldapsearch -x -b "DC=demo,DC=netapp,DC=com" ldap://192.168.0.62 -D
"CN=Manager,DC=demo,DC=netapp,DC=com" - W "objectclass=account" cn
```

### More Examples:

```
ldapsearch -b "baseDN" [ options ] -f filterFile [
attributeName ...]
```

<code>-h</code>	<i>hostname</i>	Specify the hostname of the directory server. When this option is omitted, the default is <code>localhost</code> .
<code>-p</code>	<i>port</i>	Specify the port number for accessing the directory server host. The default is 389 normally and 636 when the SSL options are used.
<code>-D</code>	<i>bindDN</i>	Specify a bind DN for accessing your directory, usually in double quotes ( " ") for the shell. If the bind DN and its password are omitted, the tool will use anonymous binding. The bind DN determines what entries and attributes will appear in the search results, according to the DN's access permissions.

-w	<i>password</i>	Specify the password for the bind DN. <b>CAUTION: Specifying the password on the command-line is a possible security risk.</b>
-w	-	Type the password for the bind DN when prompted. This is the most secure way of specifying the password.
-j	<i>filename</i>	Specify a file containing the password for the bind DN. Use this option in scripts and place the password in a secure file to protect it. This option is mutually exclusive with the -w option.
-b	<i>baseDN</i>	Specify the base DN for the search operation, usually in double quotes ( " ") for the shell. You may omit this option if you specify the base DN in the LDAP_BASEDN environment variable.
-s	<i>scope</i>	Specify the scope of a search. The <i>scope</i> parameter may have one of the following values: <ul style="list-style-type: none"> <li>• base - For searching only the base entry.</li> <li>• one - For searching only the children of the base entry.</li> <li>• sub - For searching the base entry and all its descendants. This is the default if the -s option is omitted.</li> </ul>
-f	<i>filterFile</i>	Specify the name of a file containing filter strings. This file contains one or more filters, each on a separate line; ldapsearch will perform a separate search with each filter, in the order found in the file.
-l	<i>seconds</i>	Specify the maximum number of seconds to wait for a search request to complete. Regardless of the value specified here, ldapsearch will never wait longer than is allowed by the server's nsslapd-timelimit attribute, whose default is 3,600 seconds. For more information, see “nsslapd-timelimit (Time Limit)” in <a href="#">Chapter 4 of the Sun ONE Directory Server Reference Manual</a> .
-V	<i>version</i>	Specify the LDAP protocol version number to be used for the search operation, either 2 or 3. LDAP v3 is the default; only



		specify LDAP v2 when connecting to servers that do not support v3.
-Y	<i>proxyDN</i>	Specify the proxy DN to use for the search operation, usually in double quotes ("" ) for the shell. For more information about proxy authorization, see <a href="#">Chapter 6, “Managing Access Control,”</a> in the <i>Sun ONE Directory Server Administration Guide</i> .
-a	<i>aliasMode</i>	Specify how aliases are dereferenced when encountered in a search. Note that Sun ONE Directory Server 5.2 and previous versions do not support aliases, so this option has no effect on these servers. The parameter may be one of the following values: <ul style="list-style-type: none"> <li>• <code>never</code> - Aliases are never dereferenced; this is the default.</li> <li>• <code>find</code> - Aliases are dereferenced only while finding the base DN.</li> <li>• <code>search</code> - Aliases are dereferenced when searching entries below the base DN (but not when finding the base DN).</li> <li>• <code>always</code> - Aliases are dereferenced both when finding the base DN and searching beneath it.</li> </ul>
-M		Manage smart referrals: when referrals are part of the search results, return the actual entry containing the referral instead of the entry obtained by following the referral. See <a href="#">“Creating Smart Referrals” in Chapter 2 of the</a> <i>Sun ONE Directory Server Administration Guide</i> .
-O	<i>hopLimit</i>	(Capital letter O) Specify the maximum number of referral hops to follow while searching.
-R		Specify that referrals should <i>not</i> be followed. By default, referrals are followed automatically.
-v		Verbose output mode: the tool will display additional information about the search, such as the filter string and the number of results for each search.

-n		No-op mode: use with the -v option to show what the tool would do with the given input but do not actually perform the search.
-0 (zero)		Allow runtime library version mismatches. When this option is omitted, the default behavior is to assert that the revision number of the LDAP API is greater than or equal to that used to compile the tool. Also, if the API library and the tool have the same vendor name, the tool will also assert that the vendor version number of the API is greater than or equal to that used to compile the tool. This information is based on the contents of the <code>LDAPAPIInfo</code> structure. (See the <i>Sun ONE LDAP SDK for C Programming Guide</i> .)
-H		Display the usage help text that briefly describes all options.

## How ldap search?

These are below steps used to search by ldap query

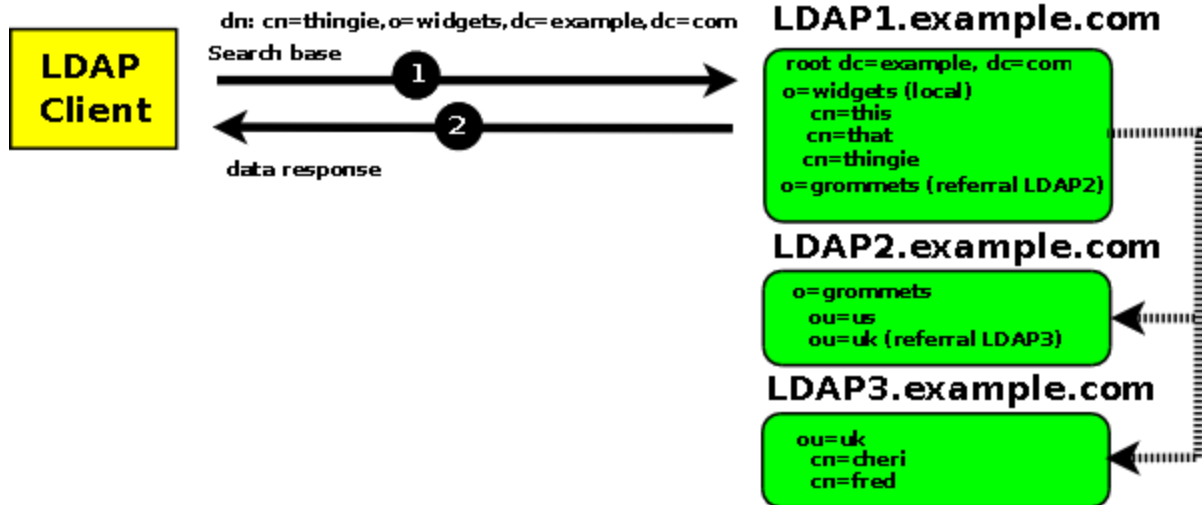
1. An ldap search for the user admin will be done by the server starting at the base dn (dc=example,dc=com).
2. When the user is found, the full dn (cn=admin,dc=example,dc=com) will be used to bind with the supplied password.
3. The ldap server will hash the password and compare with the stored hash value. If it matches, you're in.

Getting step 1 right is the hardest part. Most of the time we face problem here.

- The dn your application will use to bind to the ldap server. This happens at application startup, before any user comes to authenticate. You will have to supply a full dn, maybe something like cn=admin,dc=example,dc=com.
- The authentication method. It is usually a "simple bind".
- The user search filter. Look at the attribute named objectClass for your admin user. It will be either inetOrgPerson or user. There will be others like top, you can ignore them. In your openca configuration, there should be a string like (objectClass=inetOrgPerson). Whatever it is, make sure it matches your admin user's object Class. You can specify two object class with this search filter (|(objectClass=inetOrgPerson)(objectClass=user)).

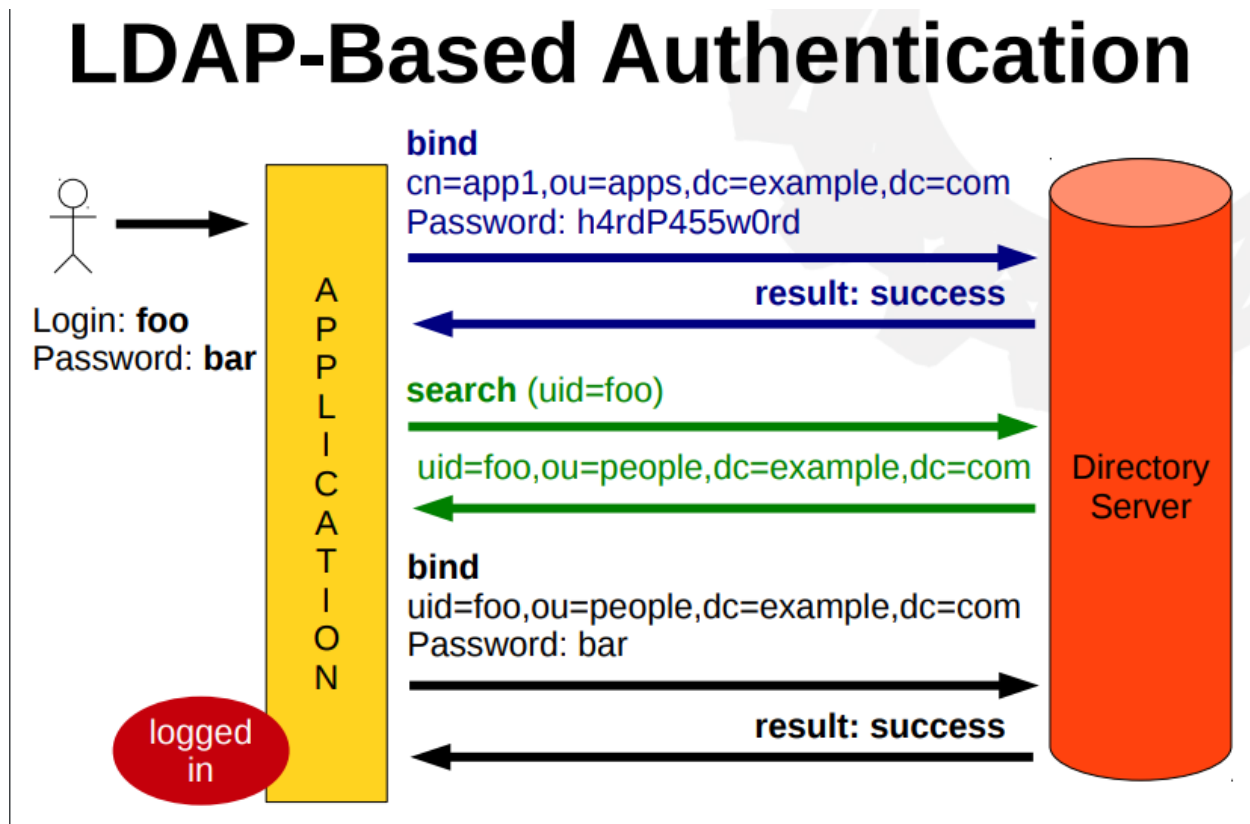
### LDAP Referrals:

Below shows a search request with a base DN of `dn:cn=thingie,o=widgets,dc=example,dc=com`, to a referral based LDAP system, that is fully satisfied from the first LDAP server (LDAP1):



Request satisfied from LDAP1 only.

## LDAP-Based Authentication



## Steps to Set Ldap in RHEL

Link: <https://www.learnitguide.net/2016/01/configure-openldap-server-on-rhel7.html>

Install & Configure Openldap Server & Client in Redhat Enterprise Linux 7:

=====

Server Configuration:

1. Install the required LDAP Packages.

```
[root@ldapserver ~]# yum -y install openldap* migrationtools
```

2. Create a LDAP root passwd for administration purpose

```
[root@ldapserver ~]# slappasswd
```

New password:

Re-enter new password:

Now save the password on text pad & It will be used later.

3. Edit the OpenLDAP Server Configuration

```
[root@ldapserver ~]# vim /etc/openldap/slapd.d/cn=config/olcDatabase={2}hdb.ldif
```

Change the olcSuffix and olcRootDN as according to your domain name

olcSuffix: dc=netapp,dc=com

olcRootDN: cn=Manager,dc=netapp,dc=com

#Add below three lines additionally in the same file at the end

olcRootPW: {SSHA}/C9pu+P5jvTbQ+6hRIopbfiC8WxyMalo --> your admin password which just now created, past it here.

olcTLSCertificateFile: /etc/pki/tls/certs/netapp.pem

olcTLSCertificateKeyFile: /etc/pki/tls/certs/netapp.pem

4. Provide the Monitor privileges.

```
[root@ldapsrvr cn=config]# vim  
/etc/openldap/slapd.d/cn=config/olcDatabase={1}monitor.ldif
```

```
vi olcDatabase={1}monitor.ldif
```

```
olcAccess: {0}to * by  
dn.base="gidNumber=0+uidNumber=0,cn=peercred,cn=external, cn=auth" read by  
dn.base="cn=Manager,dc=netapp,dc=com" read by * none
```

```
[root@ldapsrvr cn=config]# slaptest -u  
config file testing succeeded
```

5. Enable and Start the SLAPD service.

```
[root@ldapsrvr cn=config]# systemctl start slapd  
[root@ldapsrvr cn=config]# systemctl enable slapd  
[root@ldapsrvr cn=config]# netstat -lt | grep ldap
```

6. Configure the LDAP Database.

```
[root@ldapsrvr cn=config]# cp /usr/share/openldap-servers/DB_CONFIG.example  
/var/lib/ldap/DB_CONFIG
```

```
[root@ldapsrvr cn=config]# chown -R ldap:ldap /var/lib/ldap/
```

Add the following LDAP Schemas.

```
[root@ldapsrvr cn=config]# ldapadd -Y EXTERNAL -H ldapi:/// -f  
/etc/openldap/schema/cosine.ldif
```

```
[root@ldapserver cn=config]# ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/openldap/schema/nis.ldif
```

```
[root@ldapserver cn=config]# ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/openldap/schema/inetorgperson.ldif
```

## 7. Create the self-signed certificate

```
[root@linux1 cn=config]# openssl req -new -x509 -nodes -out /etc/pki/tls/certs/netapp.pem -keyout /etc/pki/tls/certs/netapp.pem -days 365
```

Country Name (2 letter code) [XX]:IN

State or Province Name (full name) []:Chennai

Locality Name (eg, city) [Default City]:Chennai

Organization Name (eg, company) [Default Company Ltd]:netapp

Organizational Unit Name (eg, section) []:DCOPS

Common Name (eg, your name or your server's hostname) []:linux1.netapp.com

Email Address []:root@linux1.netapp.com

Verify the created certificates under the location /etc/pki/tls/certs/

```
[root@ldapserver cn=config]# ll /etc/pki/tls/certs/*.pem
```

## 8. Create base objects in OpenLDAP.

```
[root@ldapserver cn=config]# cd /usr/share/migrationtools/
```

```
[root@ldapserver migrationtools]# vim migrate_common.ph
```

```
$DEFAULT_MAIL_DOMAIN = "netapp.com";
```

```
$DEFAULT_BASE = "dc=netapp,dc=com";
```

```
$EXTENDED_SCHEMA = 1;
```

## 9. Generate a base.ldif file for your Domain.

```
[root@ldapserver migrationtools]# touch /root/base.ldif
```

#### 10. Create Local Users.

```
[root@ldapserver migrationtools] # useradd ldapuser1
```

```
[root@ldapserver migrationtools] # useradd ldapuser2
```

```
[root@ldapserver migrationtools] # passwd ldapuser1
```

```
[root@ldapserver migrationtools] # passwd ldapuser2
```

```
[root@ldapserver migrationtools] # echo "redhat" | passwd --stdin ldapuser1
```

```
[root@ldapserver migrationtools] # echo "redhat" | passwd --stdin ldapuser2
```

```
[root@ldapserver migrationtools]# grep ":10[0-9][0-9]" /etc/passwd /root/passwd
```

```
[root@ldapserver migrationtools]# grep ":10[0-9][0-9]" /etc/group /root/group
```

```
[root@ldapserver migrationtools]# ./migrate_passwd.pl /root/passwd /root/users.ldif
```

```
[root@ldapserver migrationtools]# ./migrate_group.pl /root/group /root/groups.ldif
```

#### 11. Import Users in to the LDAP Database.

```
[root@ldapserver migrationtools]# ldapadd -x -W -D  
"cn=Manager,dc=netapp,dc=com" -f /root/base.ldif
```

```
[root@ldapserver migrationtools]# ldapadd -x -W -D  
"cn=Manager,dc=netapp,dc=com" -f /root/users.ldif
```

```
[root@ldapserver migrationtools]# ldapadd -x -W -D  
"cn=Manager,dc=netapp,dc=com" -f /root/groups.ldif
```

#### 12. Test the configuration.

```
[root@ldapserver migrationtools]# ldapsearch -x cn=ldapuser1 -b  
dc=netapp,dc=com
```

```
[root@ldapserver migrationtools]# ldapsearch -x -b 'dc=netapp,dc=com'  
'(objectclass=*)'
```

#### 13. Stop FirewallD to allow the connection.



```
[root@ldapserver migrationtools]# systemctl stop firewalld
```

14. NFS Configuration to export the Home Directory.

```
[root@ldapserver ~]# vim /etc/exports
```

```
/home *(rw, sync)
```

Enable and restart rpcbind and nfs service.

```
[root@ldapserver ~]# yum -y install rpcbind* nfs*
```

```
[root@ldapserver ~]# systemctl start rpcbind
```

```
[root@ldapserver ~]# systemctl start nfs
```

```
[root@ldapserver ~]# systemctl enable rpcbind
```

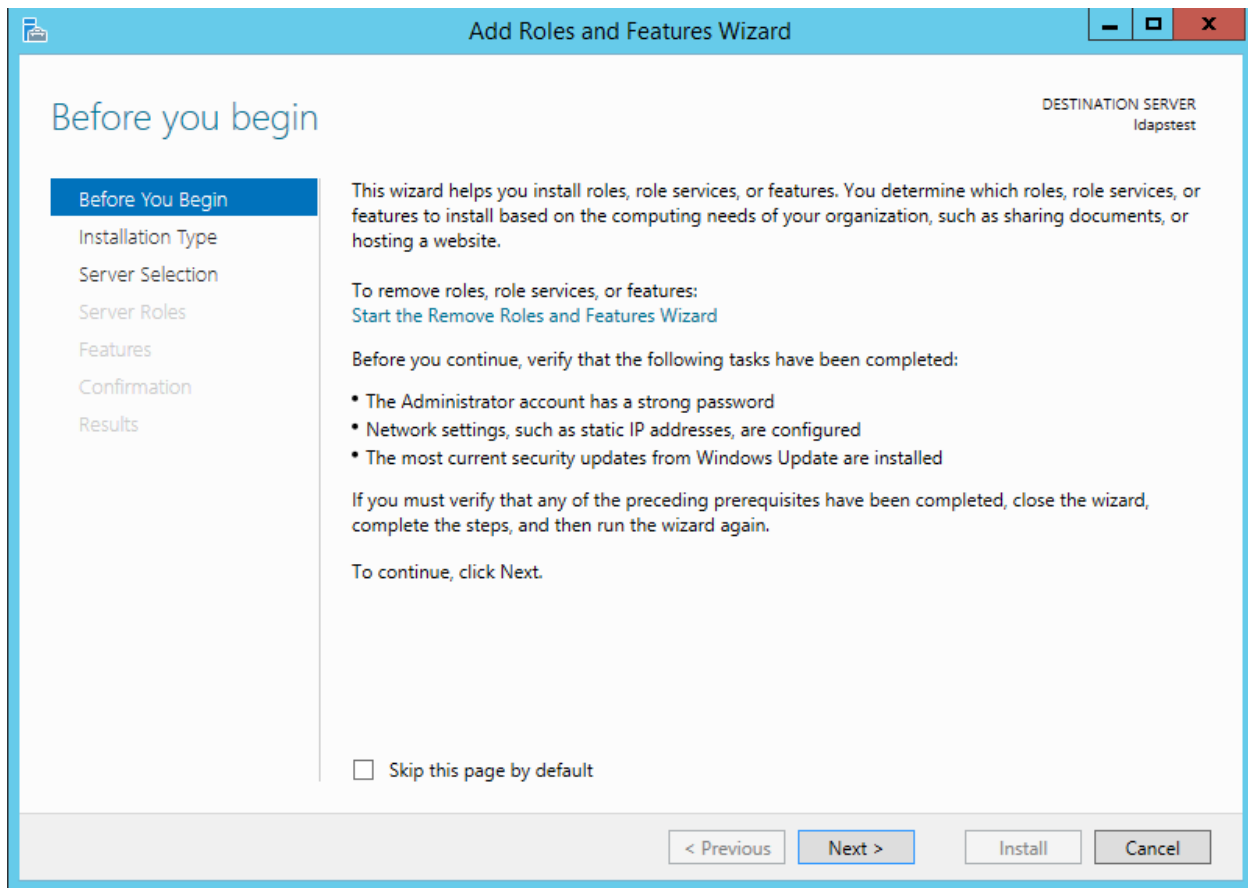
```
[root@ldapserver ~]# systemctl enable nfs
```

Test the NFS Configuration.

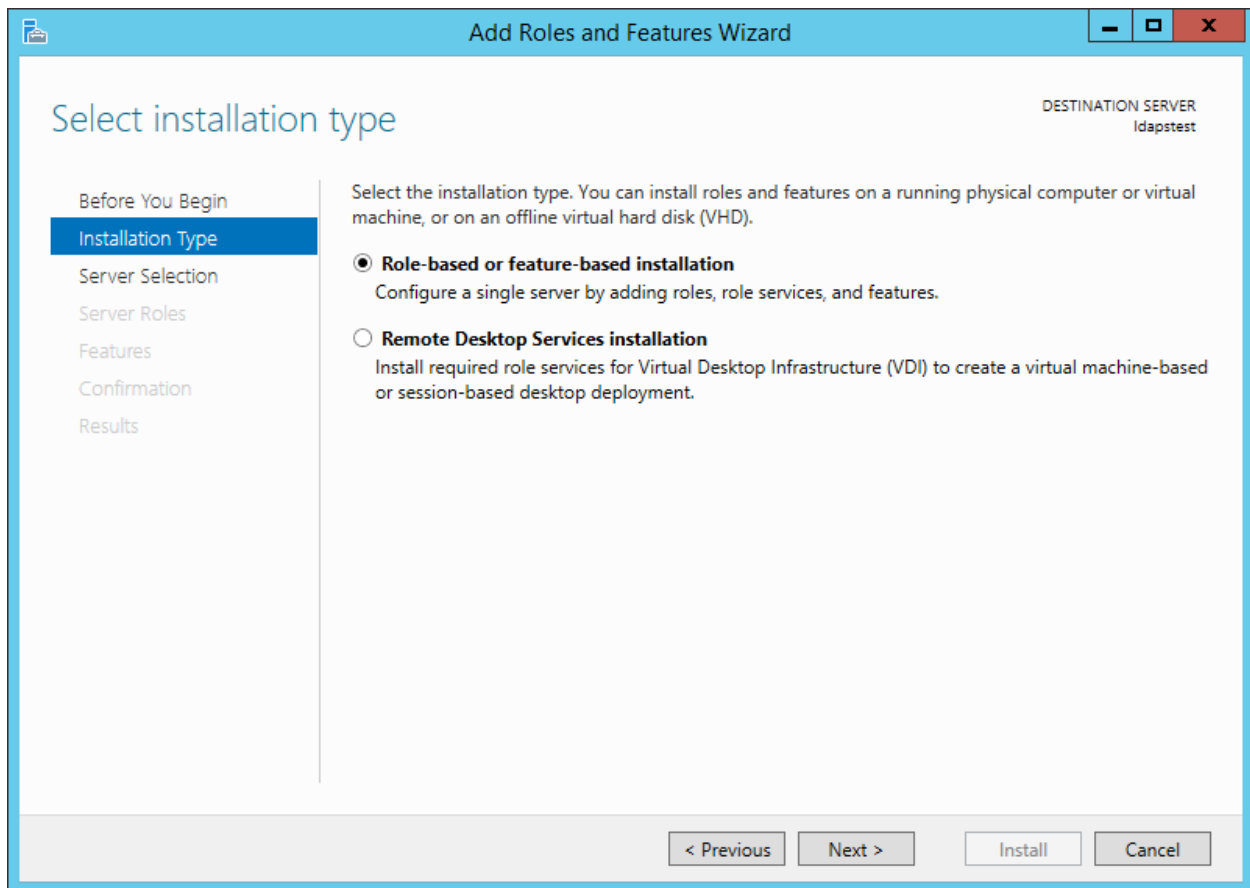
```
[root@ldapserver ~]# showmount -e
```

## Steps to setup ldap in Windows

Click on Start --> Server Manager --> Add Roles and Features. Click Next.



Choose Role-based or feature-based installation. Click Next.



Select ldapstest server from the server pool. Click Next.

Add Roles and Features Wizard

Select destination server

DESTINATION SERVER  
Idapstest

Before You Begin

Installation Type

Server Selection

Server Roles

Features

Confirmation

Results

Select a server or a virtual hard disk on which to install roles and features.

☒ Select a server from the server pool

☐ Select a virtual hard disk

Server Pool

Filter:

Name	IP Address	Operating System
Idapstest	10.5.0.4	Microsoft Windows Server 2012 R2 Datacenter

1 Computer(s) found

This page shows servers that are running Windows Server 2012, and that have been added by using the Add Servers command in Server Manager. Offline servers and newly-added servers from which data collection is still incomplete are not shown.

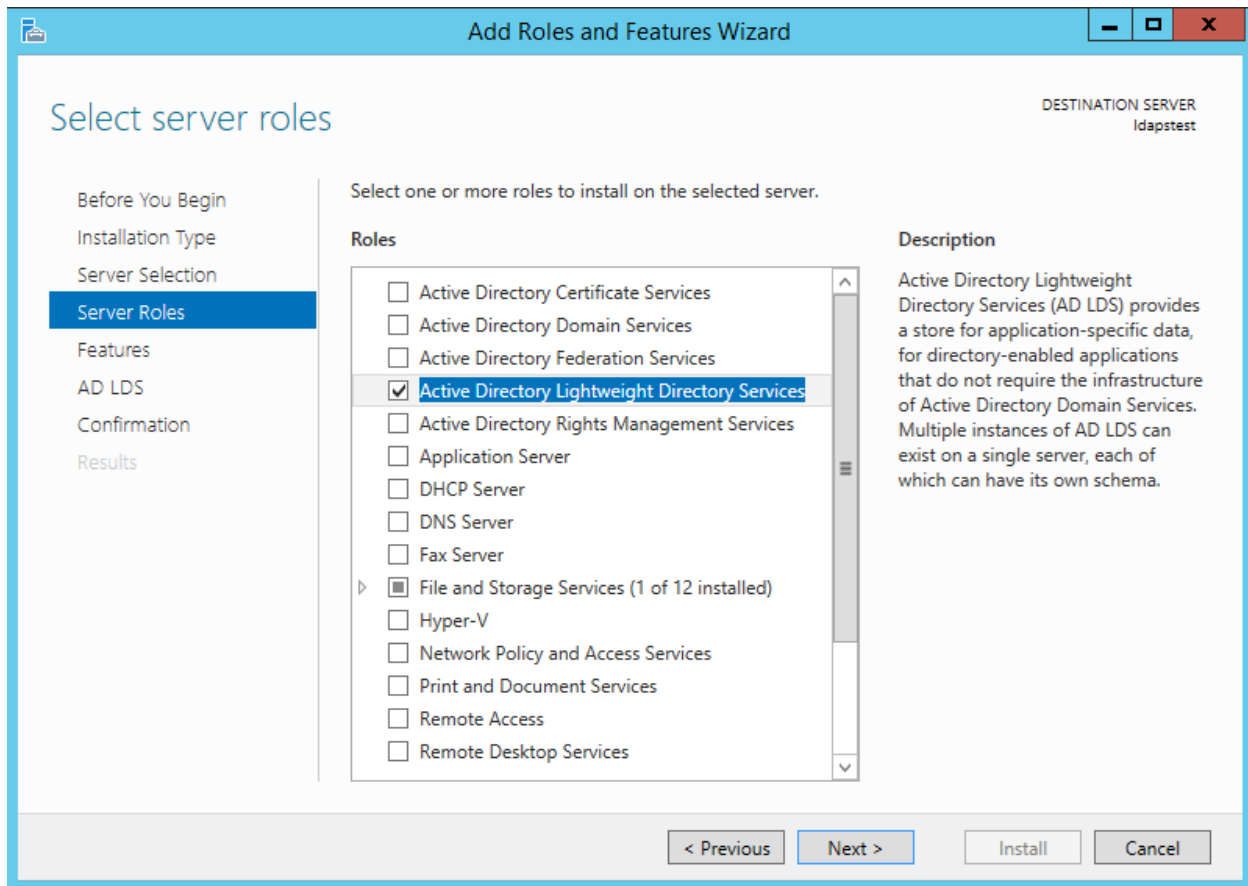
< Previous

Next >

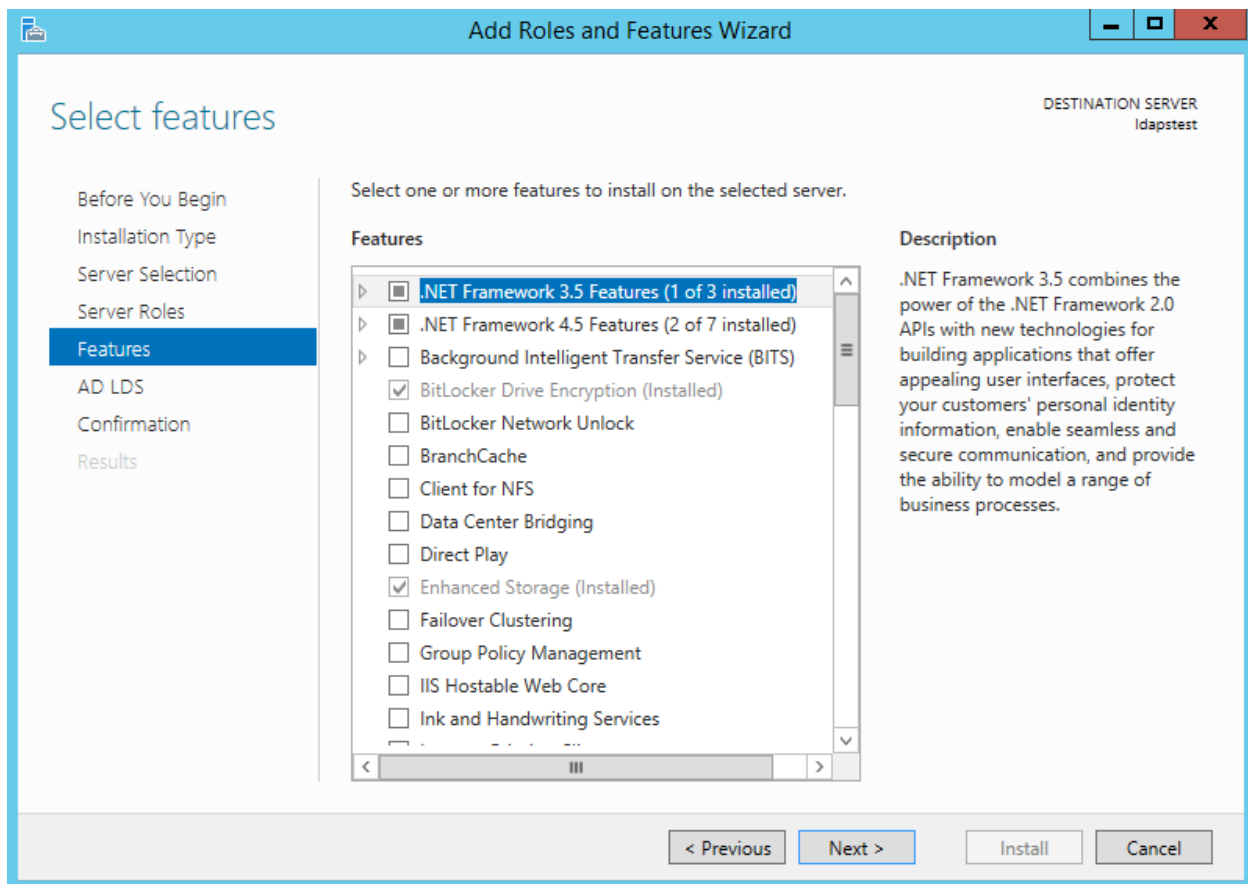
Install

Cancel

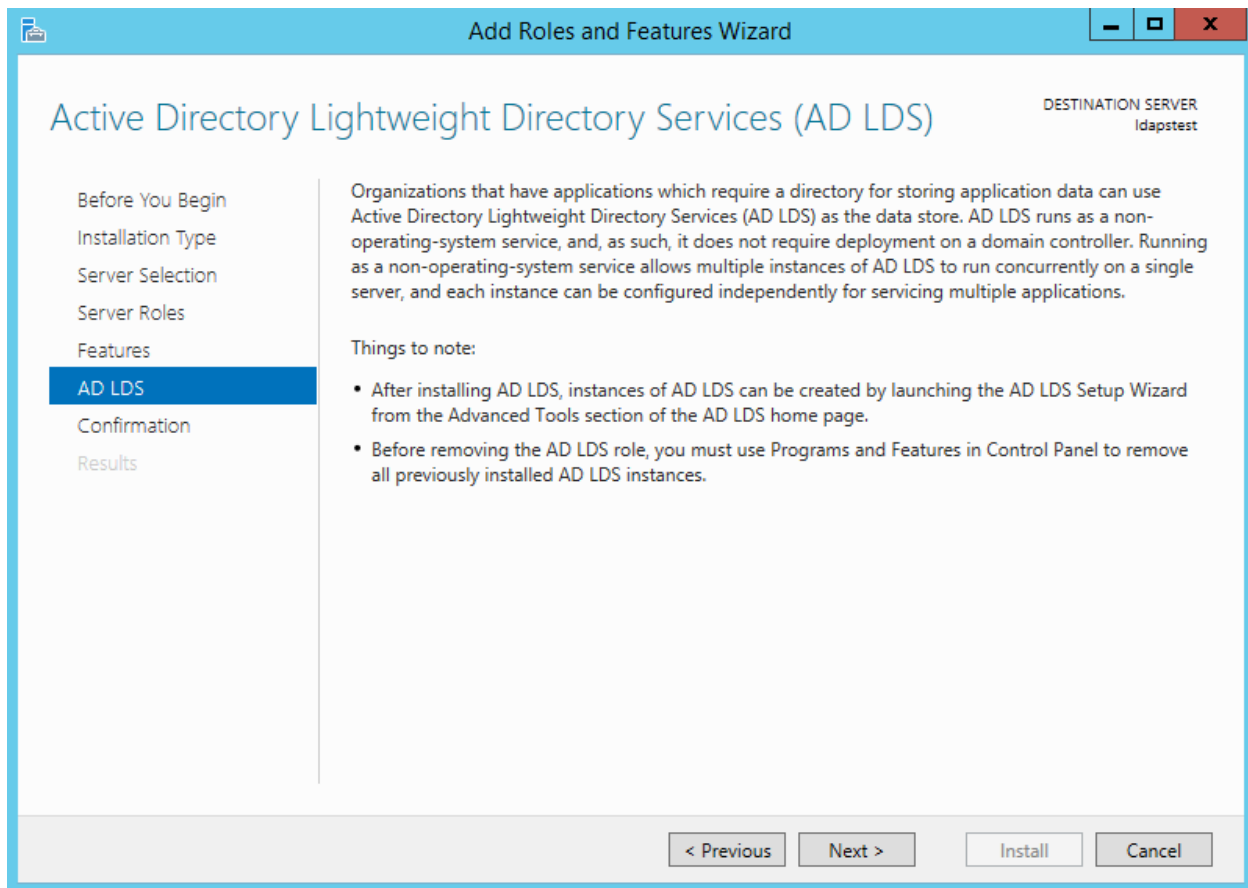
Mark Active Directory Lightweight Directory Services from the list of roles and click Next.



From the list of features, choose nothing – just click Next.

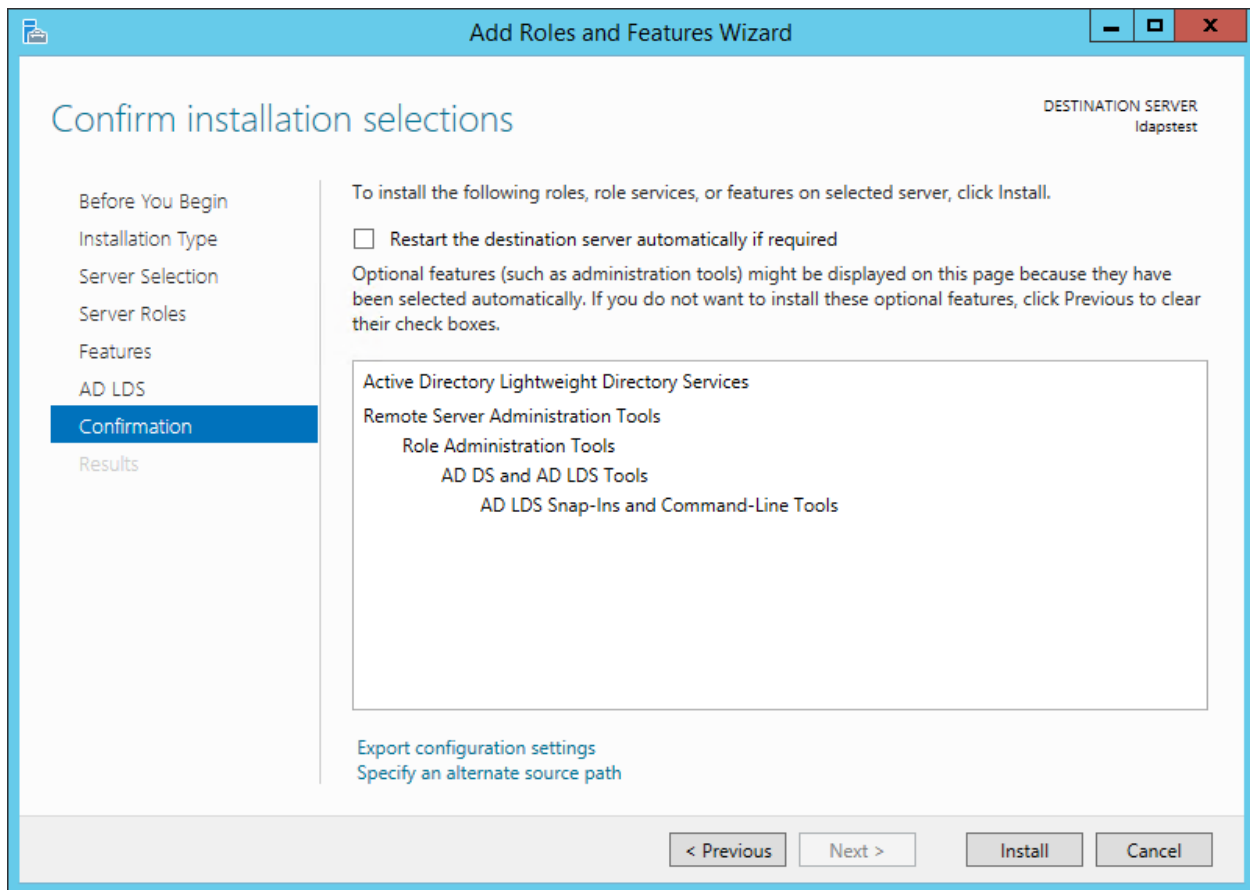


Click Next.

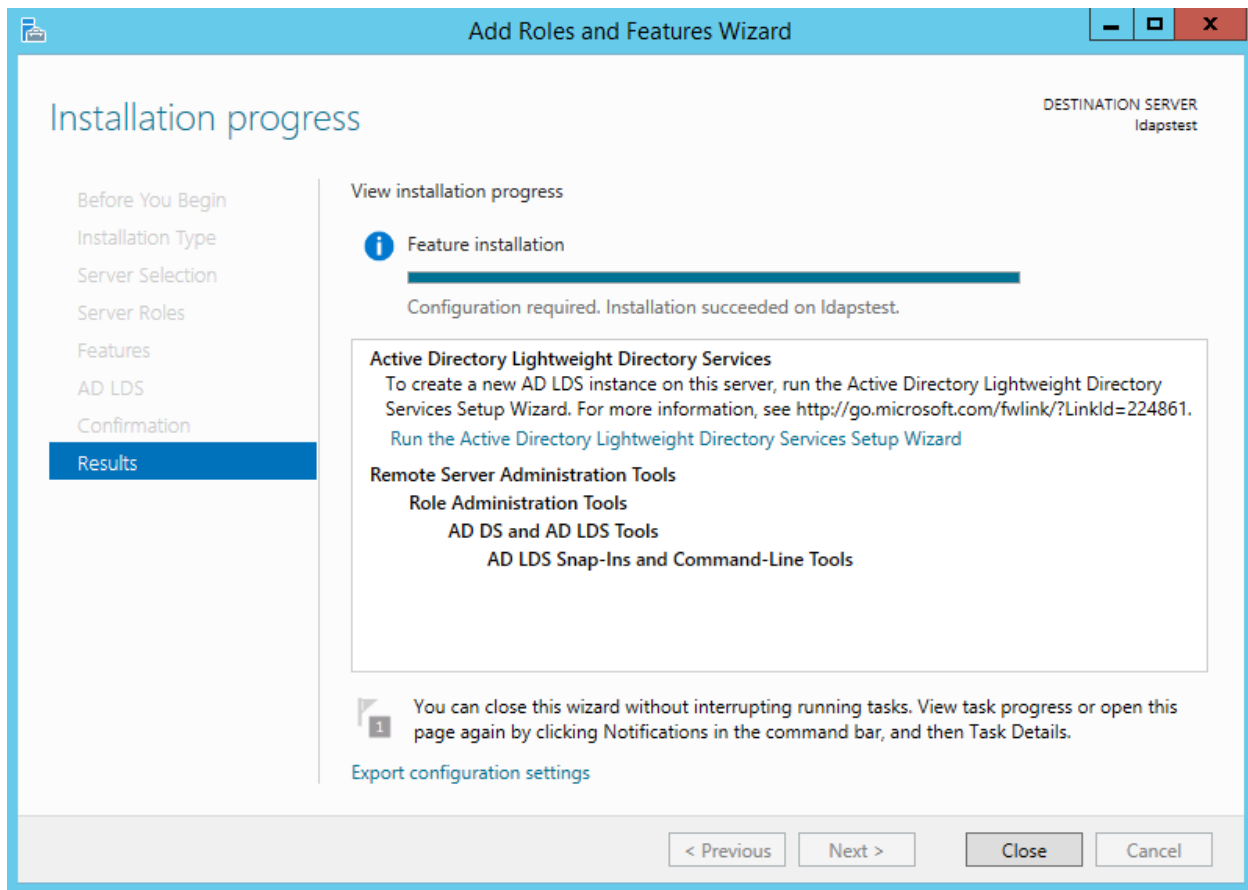


Click Install to start installation.

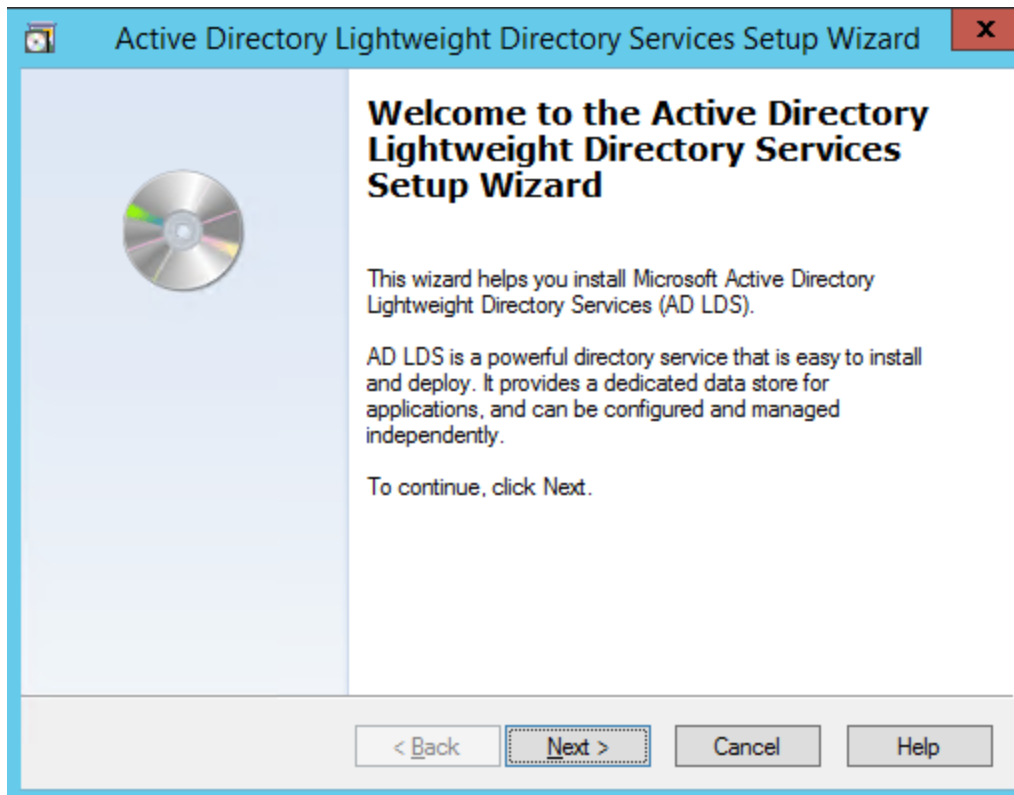




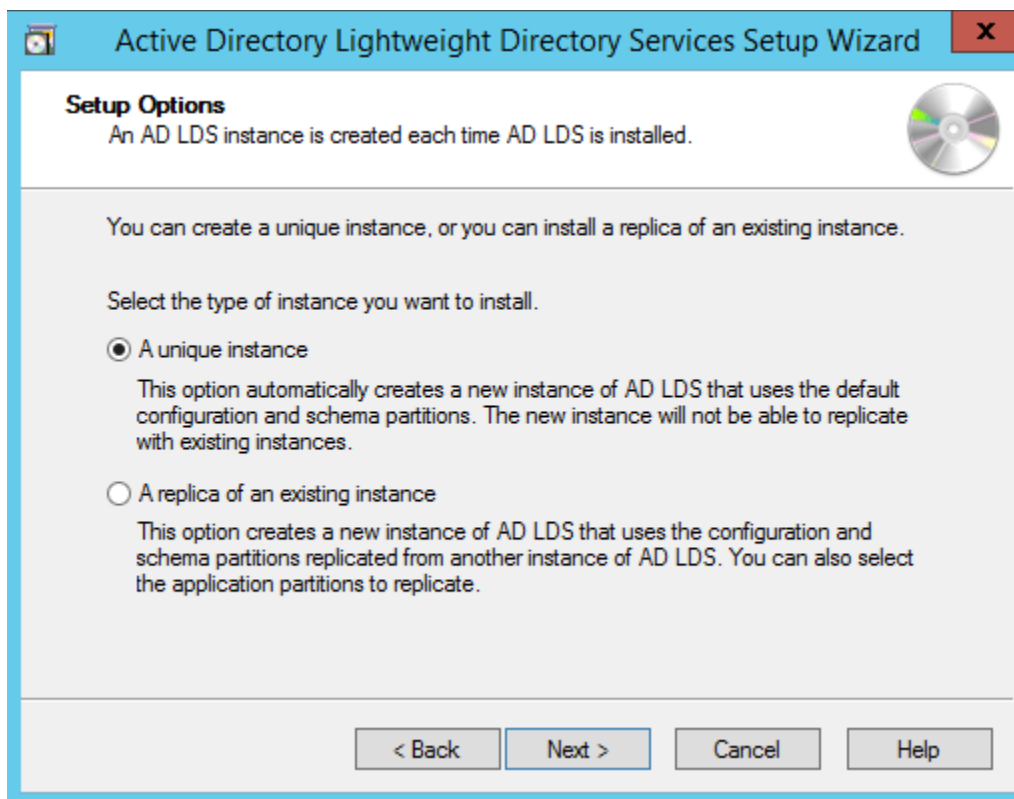
Once installation is complete, click Close.



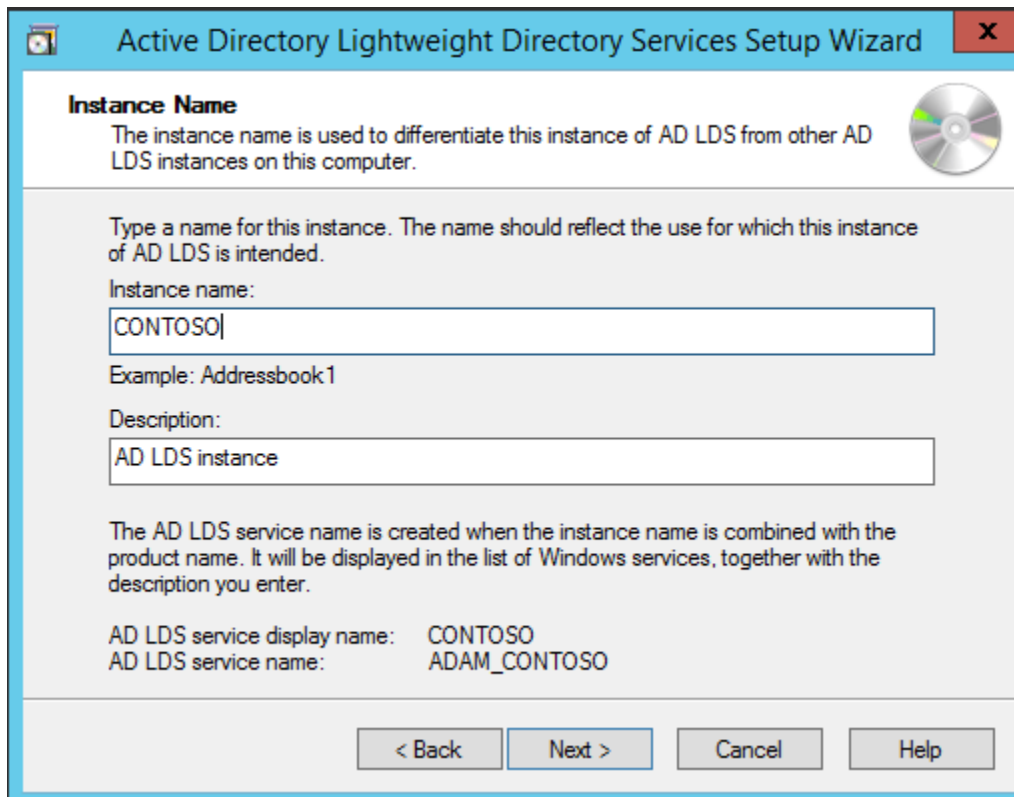
Now we have successfully set up AD LDS Role. Let us create a new AD LDS Instance “CONTOSO” using the wizard. Click the “Run the Active Directory Lightweight Directory Services Setup Wizard” in the above screen. And then Click Close.



Choose Unique Instance since we are setting it up for the first time.



Type "CONTOSO" in Instance Name and click Next.



The screenshot shows the 'Active Directory Lightweight Directory Services Setup Wizard' window. The title bar includes a close button (X). The main content area is titled 'Instance Name' and contains the following text: 'The instance name is used to differentiate this instance of AD LDS from other AD LDS instances on this computer.' Below this, there is a text box for 'Instance name' containing 'CONTOSO'. An example 'Addressbook1' is shown. A 'Description' text box contains 'AD LDS instance'. A paragraph explains that the AD LDS service name is created from the instance name and product name. At the bottom, it shows 'AD LDS service display name: CONTOSO' and 'AD LDS service name: ADAM\_CONTOSO'. Navigation buttons at the bottom are '< Back', 'Next >', 'Cancel', and 'Help'.

**Instance Name**

The instance name is used to differentiate this instance of AD LDS from other AD LDS instances on this computer.

Type a name for this instance. The name should reflect the use for which this instance of AD LDS is intended.

Instance name:

CONTOSO

Example: Addressbook1

Description:

AD LDS instance

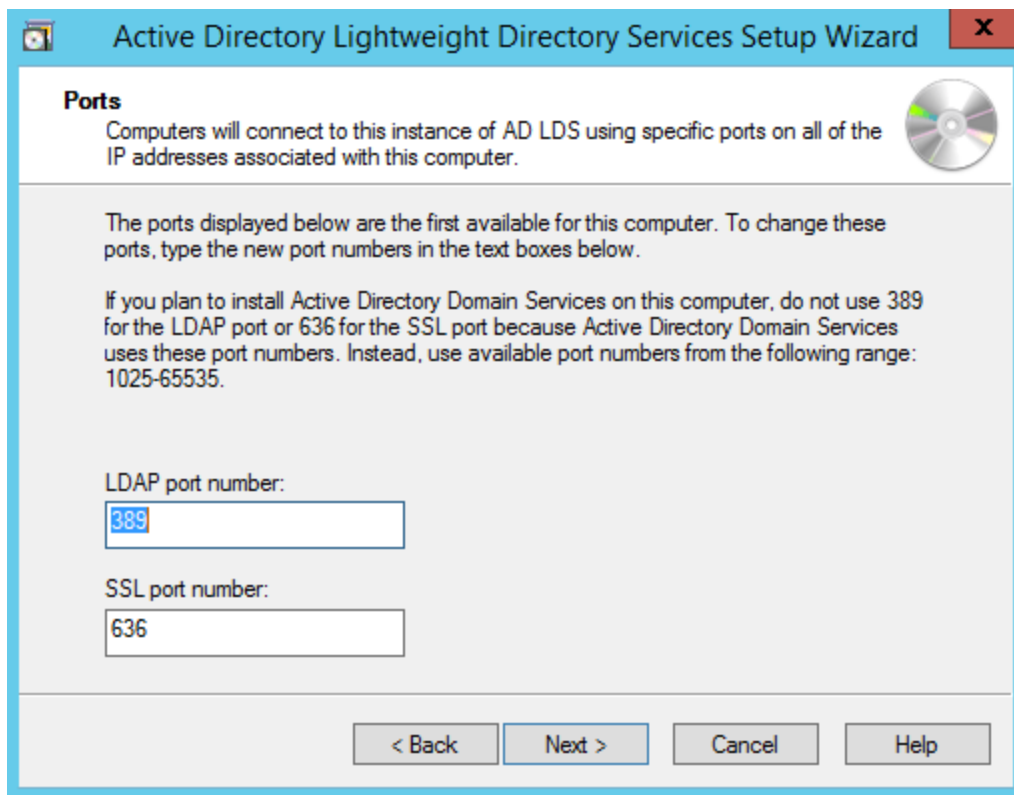
The AD LDS service name is created when the instance name is combined with the product name. It will be displayed in the list of Windows services, together with the description you enter.

AD LDS service display name: CONTOSO  
AD LDS service name: ADAM\_CONTOSO

< Back   Next >   Cancel   Help

**Note:** You can use any name instead of CONTOSO

By Default, LDAP Port is 389 and LDAPS port is 636, let us choose the default values - click Next.



The screenshot shows the 'Ports' screen of the Active Directory Lightweight Directory Services Setup Wizard. The title bar reads 'Active Directory Lightweight Directory Services Setup Wizard'. The main heading is 'Ports'. Below it, a paragraph states: 'Computers will connect to this instance of AD LDS using specific ports on all of the IP addresses associated with this computer.' To the right of this text is a CD icon. Another paragraph explains: 'The ports displayed below are the first available for this computer. To change these ports, type the new port numbers in the text boxes below.' A third paragraph provides a warning: 'If you plan to install Active Directory Domain Services on this computer, do not use 389 for the LDAP port or 636 for the SSL port because Active Directory Domain Services uses these port numbers. Instead, use available port numbers from the following range: 1025-65535.' Below this, there are two text input fields. The first is labeled 'LDAP port number:' and contains the value '389'. The second is labeled 'SSL port number:' and contains the value '636'. At the bottom of the window are four buttons: '< Back', 'Next >', 'Cancel', and 'Help'.

**Ports**

Computers will connect to this instance of AD LDS using specific ports on all of the IP addresses associated with this computer.

The ports displayed below are the first available for this computer. To change these ports, type the new port numbers in the text boxes below.

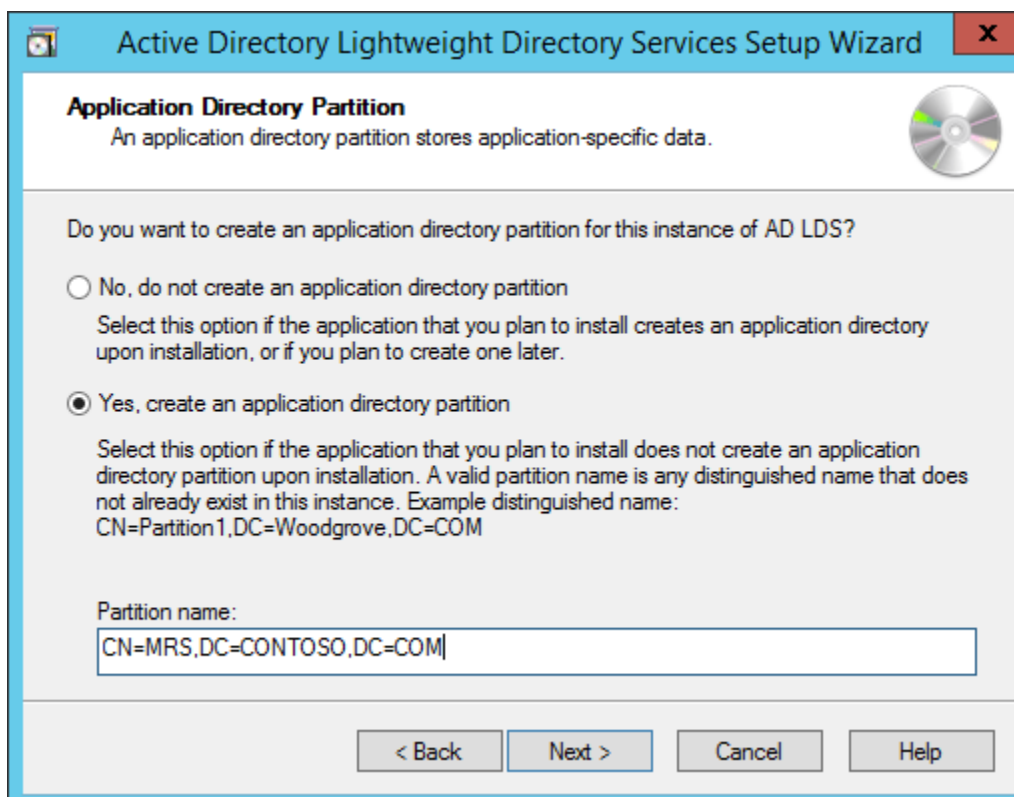
If you plan to install Active Directory Domain Services on this computer, do not use 389 for the LDAP port or 636 for the SSL port because Active Directory Domain Services uses these port numbers. Instead, use available port numbers from the following range: 1025-65535.

LDAP port number:  
389

SSL port number:  
636

< Back   Next >   Cancel   Help

Create a new Application Directory Partition named "CN=MRS,DC=CONTOSO,DC=COM". Click Next.



The screenshot shows the 'Application Directory Partition' screen of the Active Directory Lightweight Directory Services Setup Wizard. The title bar reads 'Active Directory Lightweight Directory Services Setup Wizard'. The main heading is 'Application Directory Partition'. Below it, a paragraph states: 'An application directory partition stores application-specific data.' To the right of this text is a CD icon. The main question is 'Do you want to create an application directory partition for this instance of AD LDS?'. There are two radio button options. The first is 'No, do not create an application directory partition', with a subtext: 'Select this option if the application that you plan to install creates an application directory upon installation, or if you plan to create one later.' The second option is selected: 'Yes, create an application directory partition', with a subtext: 'Select this option if the application that you plan to install does not create an application directory partition upon installation. A valid partition name is any distinguished name that does not already exist in this instance. Example distinguished name: CN=Partition1,DC=Woodgrove,DC=COM'. Below this, there is a text input field labeled 'Partition name:' which contains the value 'CN=MRS,DC=CONTOSO,DC=COM'. At the bottom of the window are four buttons: '< Back', 'Next >', 'Cancel', and 'Help'.

**Application Directory Partition**

An application directory partition stores application-specific data.

Do you want to create an application directory partition for this instance of AD LDS?

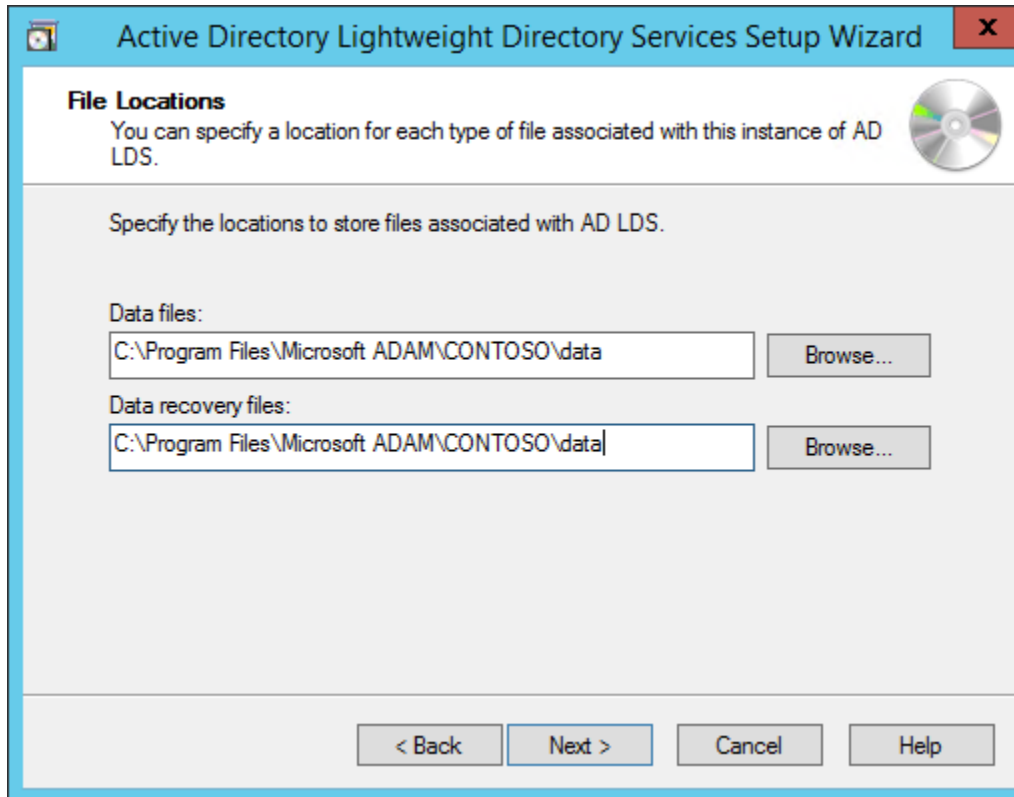
☐ No, do not create an application directory partition  
Select this option if the application that you plan to install creates an application directory upon installation, or if you plan to create one later.

☒ Yes, create an application directory partition  
Select this option if the application that you plan to install does not create an application directory partition upon installation. A valid partition name is any distinguished name that does not already exist in this instance. Example distinguished name:  
CN=Partition1,DC=Woodgrove,DC=COM

Partition name:  
CN=MRS,DC=CONTOSO,DC=COM

< Back   Next >   Cancel   Help

Using the default values for storage location of AD LDS files- Click Next.



The image shows a screenshot of the 'Active Directory Lightweight Directory Services Setup Wizard' window. The title bar is blue with a yellow icon on the left and a red close button on the right. The main window has a light blue header bar with the title 'Active Directory Lightweight Directory Services Setup Wizard'. Below the header, the 'File Locations' section is highlighted in white. It contains the text 'You can specify a location for each type of file associated with this instance of AD LDS.' and a CD icon. The main area is light gray and contains the instruction 'Specify the locations to store files associated with AD LDS.' Below this, there are two sections: 'Data files:' and 'Data recovery files:'. Each section has a text box containing the default path 'C:\Program Files\Microsoft ADAM\CONTOSO\data' and a 'Browse...' button. At the bottom of the window, there are four buttons: '< Back', 'Next >', 'Cancel', and 'Help'.

**File Locations**  
You can specify a location for each type of file associated with this instance of AD LDS.

Specify the locations to store files associated with AD LDS.

Data files:  
C:\Program Files\Microsoft ADAM\CONTOSO\data Browse...

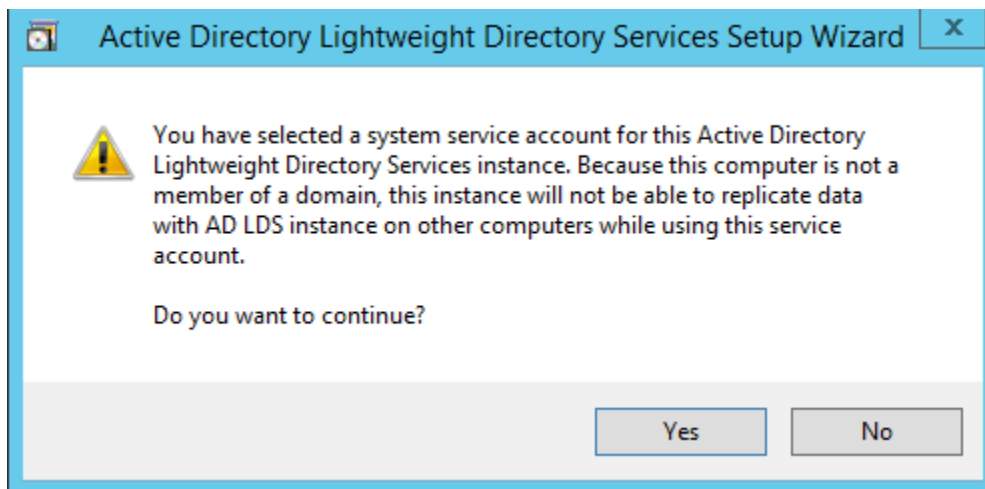
Data recovery files:  
C:\Program Files\Microsoft ADAM\CONTOSO\data Browse...

< Back Next > Cancel Help

Choosing Network Service Account for running the AD LDS Service.



You will receive a prompt warning about data replication. Since we are using a single LDAP Server, we can click Yes.



Choosing the currently logged on user as an administrator for the AD LDS Instance. Click Next.

**Active Directory Lightweight Directory Services Setup Wizard**

### AD LDS Administrators

You can specify the user or group that will have administrative privileges for this instance of AD LDS.

Assign the following user or group of users administrative permissions for AD LDS.

☒ Currently logged on user: DEPLOYRLDAPS\azureuser  
The user that is installing AD LDS will have administrative permissions for this instance of AD LDS.

☐ This account  
The selected user or group will have administrative permissions for this instance of AD LDS. You can choose any user or group from this computer, this computer's domain, or any domain that is trusted by this computer's domain.

Account name:

< Back   Next >   Cancel   Help

Mark all the required LDIF files to import (Here we are marking all files). Click Next.

**Active Directory Lightweight Directory Services Setup Wizard**

### Importing LDIF Files

You can import data from Lightweight Directory Interchange Format (LDIF) files into your AD LDS application directory partition.

To configure the AD LDS service in a specific way, import one or more of the LDIF files listed below.

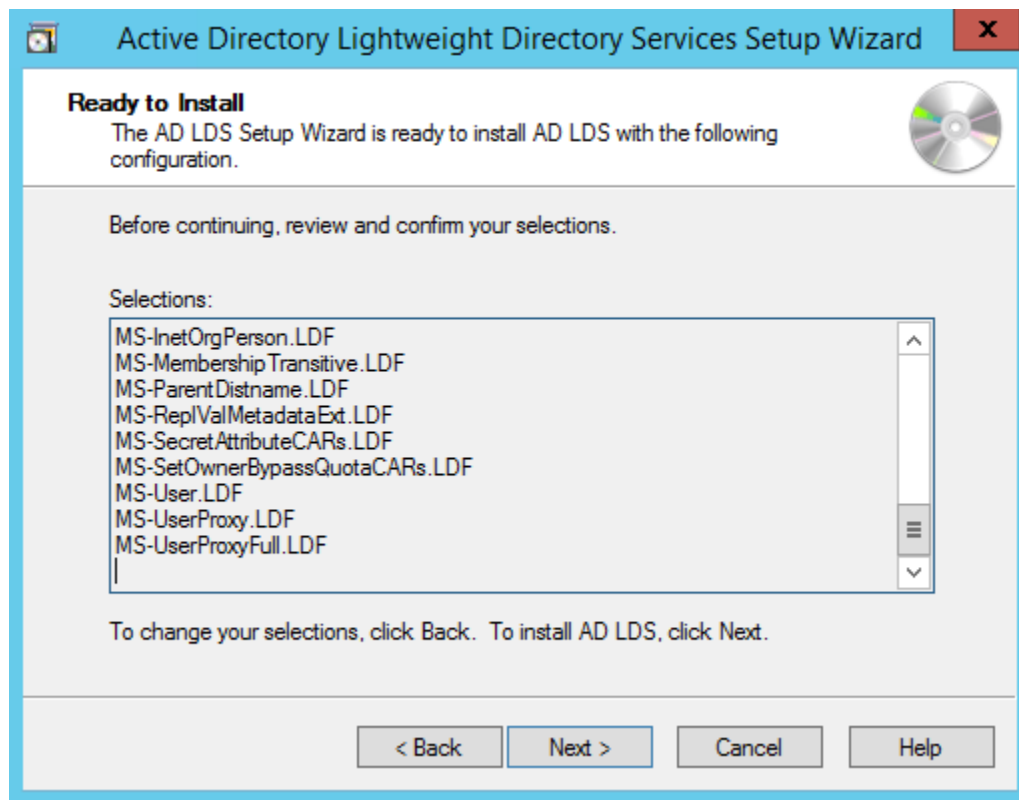
LDIF file name	Description
<input checked="" type="checkbox"/> MS-AdamSyncMetadata.LDF	ADAMSync metadata schema extension. Required for...
<input checked="" type="checkbox"/> MS-ADLDS-DisplaySpecifiers.L...	AD LDS Display specifiers schema and display specifi...
<input checked="" type="checkbox"/> MS-AZMan.LDF	AD LDS schema extensions for AzMan.
<input checked="" type="checkbox"/> MS-InetOrgPerson.LDF	AD LDS inetOrgPerson, user and related classes.
<input checked="" type="checkbox"/> MS-MembershipTransitive.LDF	AD LDS membership transitive.
<input checked="" type="checkbox"/> MS-ParentDistname.LDF	AD LDS parent dist name.
<input checked="" type="checkbox"/> MS-ReplValMetadataExt.LDF	AD LDS ReplValueMetaDataSet.
<input checked="" type="checkbox"/> MS-SecretAttributeCABs.LDF	AD LDS Secret Attribute Control Access Rights

<   |||   >

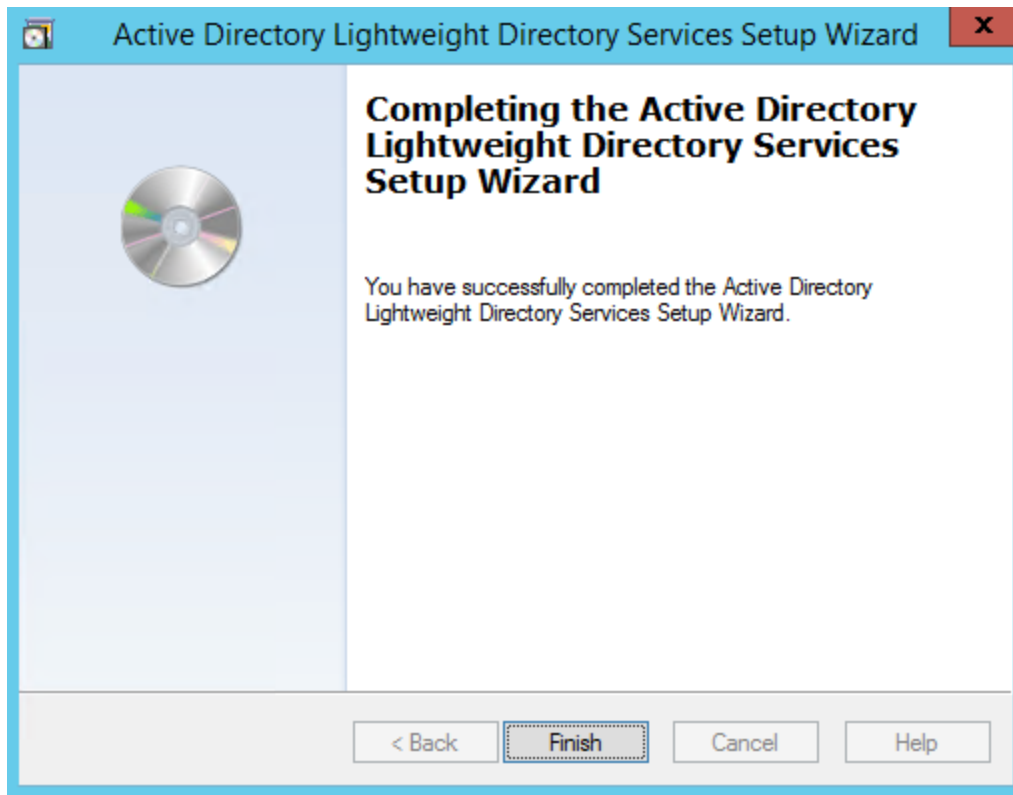
< Back   Next >   Cancel   Help



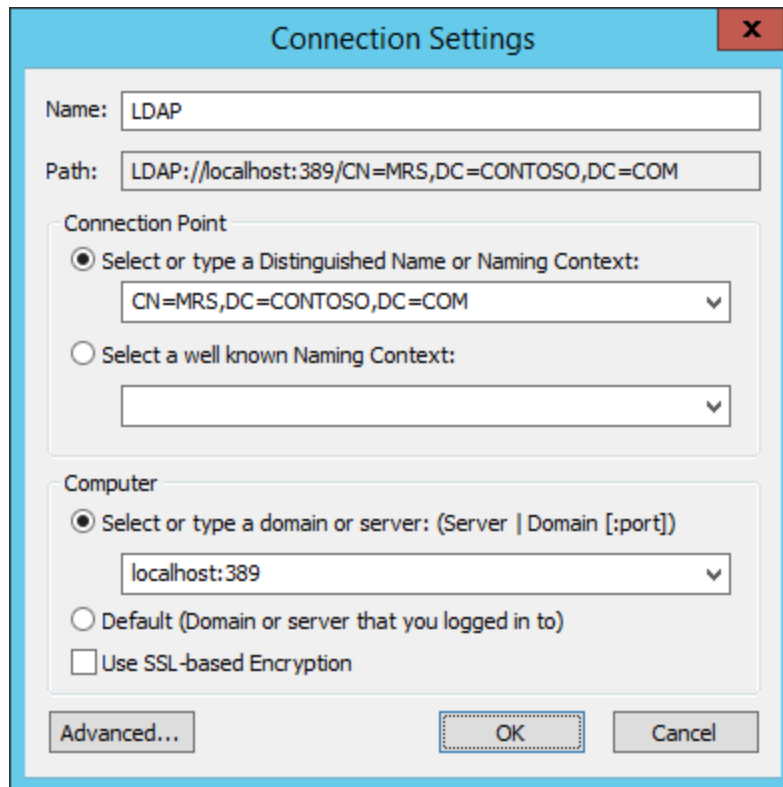
Verify that all the selections are right and then Click Next to confirm Installation.



Once the instance is setup successfully, click Finish.



Now let us try to connect to the AD LDS Instance CONTOSO using ADSI Edit.  
Click on Start --> Search "ADSI Edit" and open it.  
Right Click on ADSI Edit Folder (on the left pane) and choose Connect To.. . Fill the following values and Click OK.

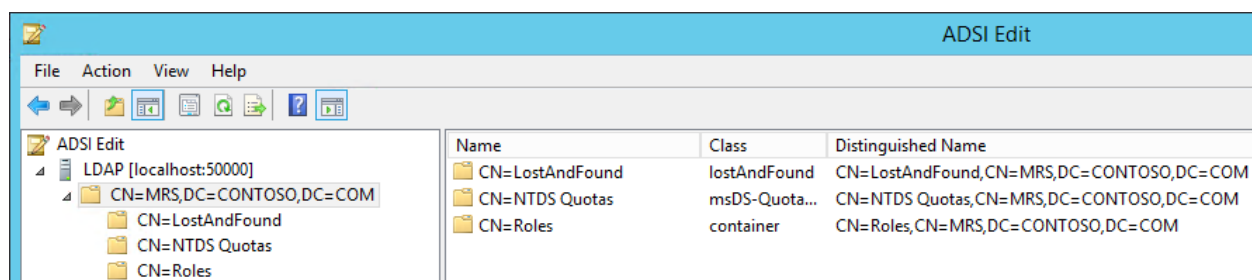


The Connection Settings dialog box is shown with the following configuration:

- Name:** LDAP
- Path:** LDAP://localhost:389/CN=MRS,DC=CONTOSO,DC=COM
- Connection Point:**
  - ☒ Select or type a Distinguished Name or Naming Context:
    - CN=MRS,DC=CONTOSO,DC=COM
  - ☐ Select a well known Naming Context:
- Computer:**
  - ☒ Select or type a domain or server: (Server | Domain [:port])
    - localhost:389
  - ☐ Default (Domain or server that you logged in to)
  - ☐ Use SSL-based Encryption

Buttons at the bottom: Advanced..., OK, Cancel

If the connection is successful, we will be able to browse the Directory  
CN=MRS,DC=CONTOSO,DC=COM :



The ADSI Edit window displays the directory structure for the LDAP connection. The left pane shows the tree view, and the right pane shows a table of objects.

Name	Class	Distinguished Name
CN=LostAndFound	lostAndFound	CN=LostAndFound,CN=MRS,DC=CONTOSO,DC=COM
CN=NTDS Quotas	msDS-Quota...	CN=NTDS Quotas,CN=MRS,DC=CONTOSO,DC=COM
CN=Roles	container	CN=Roles,CN=MRS,DC=CONTOSO,DC=COM

How to configure DFM:

dfm ldap add ~~172.19.233.90~~

dfm ldap template netscape

dfm options set ldapEnabled=Yes

dfm option set ldapBaseDN="dc=netapp,dc=local"

dfm option set ldapBindDN="CN=test user (contractor),OU=test,DC=netapp,DC=local"

dfm options set ldapBindPass=mohan@143

dfm options set ldapGID=memberOf

dfm options set ldapMember=member

dfm options set ldapUGID=CN

dfm options set ldapUID=sAMAccountName

dfm options set ldapVersion=3

#dfm ldap test pkandru test@123

Authentication succeeded.

Username: CN=pradeep Kandru(Contractor - (M)Manger,OU=test,DC=netapp,DC=local

Name: CN=pradeep Kandru(Contractor - (M)Manger,OU=test,DC=netapp,DC=local

Name: memberOf=CN=Administrators,CN=Builtin,DC=netapp,DC=local

Name: CN=Administrators,CN=Builtin,DC=netapp,DC=local

DFM community links:

1. <https://community.netapp.com/t5/Data-Infrastructure-Management-Software-Discussions/QUESTION-Authenticating-OnCMD-with-W2K8-AD-on-a-RHEL-box/td-p/28680>
2. <https://www.openldap.org/doc/admin23/quickstart.html>
3. <https://library.netapp.com/ecmdocs/ECMP1608418/html/GUID-F4F1CB5D-A757-48F6-975F-1237416D652A.html>
4. <https://community.netapp.com/t5/Data-Infrastructure-Management-Software-Discussions/DFM-Linux-and-Active-Directory-service-account/td-p/41064>
5. <https://www.ontap8.com/ldapauthenticationocum-on-linux/>
6. <http://www.cosonok.com/2016/12/ocum-7-dfm-cli-reference.html>
7. <http://www.cosonok.com/2017/07/ocum-72-dfm-cli-and-um-cli-reference.html>