Understanding LDAP v3 namespace configuration in Cognos

This document explains how to configure Cognos to use any LDAP v3 compliant directory server as a Cognos Namespace.

Any LDAP browser is strongly recommended, to understand LDAP structure and to follow below steps. In this example Apache Directory Studio has been used.

LDAP namespace configuration parameters in Cognos:

Main settings

Namespace ID

An unique ID used to identify the namespace. Any unique string is allowed.

Host and port

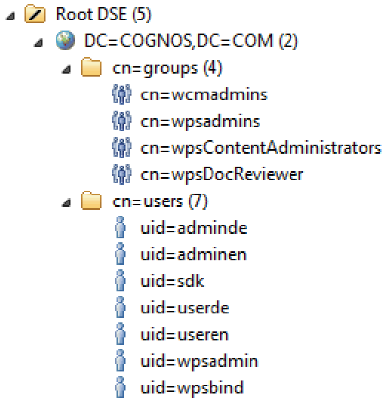
Specifies the the host name / IP and port for the LDAP directory server.

Base Distinguished Name

It is the starting place for all searches Cognos will do. Keep sure all required users, groups and folders belongs to this path.

If it is set for example to *cn=users,dc=cognos,dc=com* then Cognos will display objects from this path only, while others, like *cn=groups,dc=cognos,dc=com* or *cn=root,dc=cognos,dc=com* will not be visible.
On the other hand it is not recommended to set too wide Base Distinguished Name like root path *dc=com* as then all searches will take more time and affect both LDAP and Cognos performance.

Here is how it does look like in a LDAP browser:



```
Root DSE (5)
  DC=COGNOS,DC=COM (2)
    cn=groups (4)
      cn=wcmadmins
      cn=wpsadmins
      cn=wpsContentAdministrators
      cn=wpsDocReviewer
    cn=users (7)
      uid=adminde
      uid=adminen
      uid=sdk
      uid=userde
      uid=useren
      uid=wpsadmin
      uid=wpsbind
```

So, for example, based on the structure above: To be able to see both users and groups Base Distinguished Name has to be set to *dc=cognos,dc=com***.**
Using *cn=users,dc=cognos,dc=com* as Base Distinguished Name we will be able to see users, but not groups. Using *cn=groups,dc=cognos,dc=com* as Base Distinguished Name we will be able to see groups but not users. Also no one may be able to login to Cognos.

User lookup
This property specifies the string used to search for user in LDAP directory. All instances of *${userID}* will be replaced by the value typed in by the user at the logon.

If the string does not begin with an open parenthesis, the result of the substitution is assumed to be a DN which can be used for authentication.

Example:
- user lookup is *uid=${userID},ou=people,dc=groups,dc=cognos,dc=com*
In this case if user will type *johndoe* Cognos will directly try to authenticate the user as:
uid=johndoe,ou=people,dc=groups,dc=cognos,dc=com
This option is recommended for case when all users have the same DN path.
The advantage is, that no time consuming LDAP search is required.

If the user lookup string does begin with an open parenthesis, the result of the substitution is assumed to be a LDAP search filter. Cognos will use the search filter to find the user account, and then will try to authenticate the user (if found) with password provided during logon.
This LDAP search string must meet two restrictions:
- conform RFC 1960 standard for LDAP search string, http://www.ietf.org/rfc/rfc1960.txt, see also
- return not more than one account as a result of search

Example 1:

- user typed *useren* as his login
- user lookup is *(uid=${userID})*
- base distinguished name is *cn=users,dc=cognos,dc=com*

In this case Cognos will search for:
- accounts inside path *cn=users,dc=cognos,dc=com*
- where attribute *uid=useren*

Example 2:

- user typed *useren@cognos.com* as his login
- user lookup is *(&(mail=${userID})(objectclass=person))*
- base distinguished name is *cn=users,dc=cognos,dc=com*

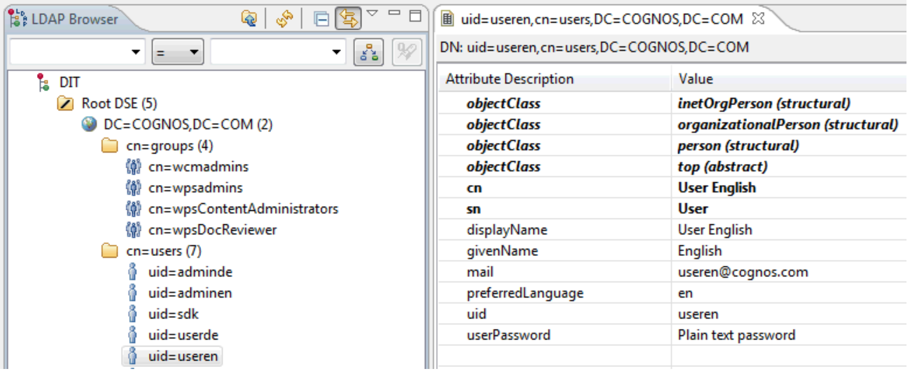
In this case Cognos will search for:
- all LDAP objects inside *cn=users,dc=cognos,dc=com*
- where attribute *mail=useren@cognos.com* and where *objectclass=person*

Example 3:

- user typed *useren@cognos.com* or *useren* as his login
- user lookup is *(|(mail=${userID})(uid=${userID}))*
- base distinguished name is *cn=users,dc=cognos,dc=com*

In this case Cognos will search for:
- all LDAP objects inside *cn=users,dc=cognos,dc=com*
- where attribute *mail* or where attribute *uid* match what the user has typed in (both his login and email is correct)

Below screen explains how it does look like in LDAP:



```
DIT
  Root DSE (5)
    DC=COGNOS,DC=COM (2)
      cn=groups (4)
        cn=wcmadmins
        cn=wpsadmins
        cn=wpsContentAdministrators
        cn=wpsDocReviewer
      cn=users (7)
        uid=adminde
        uid=adminen
        uid=sdk
        uid=userde
        uid=useren
```

uid=useren,cn=users,DC=COGNOS,DC=COM

DN: uid=useren,cn=users,DC=COGNOS,DC=COM

Attribute Description	Value
objectClass	***inetOrgPerson (structural)***
objectClass	***organizationalPerson (structural)***
objectClass	***person (structural)***
objectClass	***top (abstract)***
cn	**User English**
sn	**User**
displayName	User English
givenName	English
mail	useren@cognos.com
preferredLanguage	en
uid	useren
userPassword	Plain text password

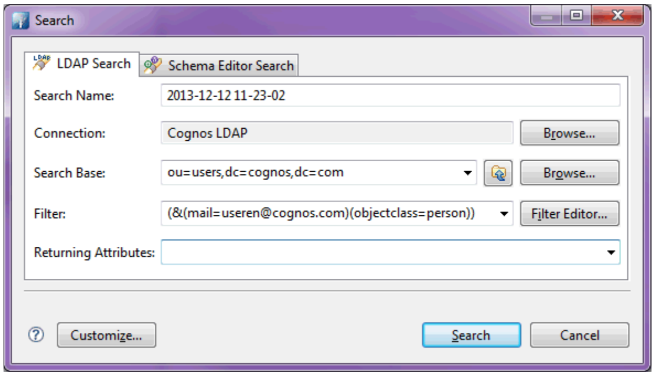
Here, we have an user identified by *uid=useren*, who has attributes (right pane):
objescClass=person
uid=useren
mail =useren@cognos.com

We used all of them in both examples above. Any other attribute can be used as well, if only it has unique value for every user to define them unambiguously.

If a LDAP namespace can be successfully tested in Cognos Configuration, but users are unable to log in then very likely user lookup string has to be carefully reviewed, a LDAP browser can be very useful to test the search string designed for Cognos namespace.

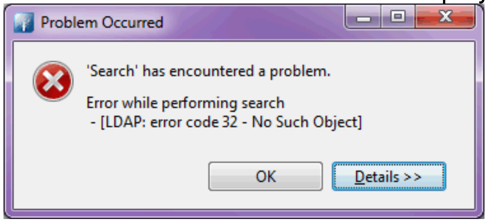
For example let's focus on a case:
- user typed *useren@cognos.com* as his login
- user lookup is *(&(mail=${userID})(objectclass=person))*
- base distinguished name is *ou=users,dc=cognos,dc=com*

Based on this we a search has been run in a LDAP browser as below:
- search string (filter) is *(&(mail=useren@cognos.com)(objectclass=person))*
- search base distinguished name is *ou=users,dc=cognos,dc=com*



Search Name: 2013-12-12 11-23-02
Connection: Cognos LDAP
Search Base: ou=users,dc=cognos,dc=com
Filter: (&(mail=useren@cognos.com)(objectclass=person))
Returning Attributes:

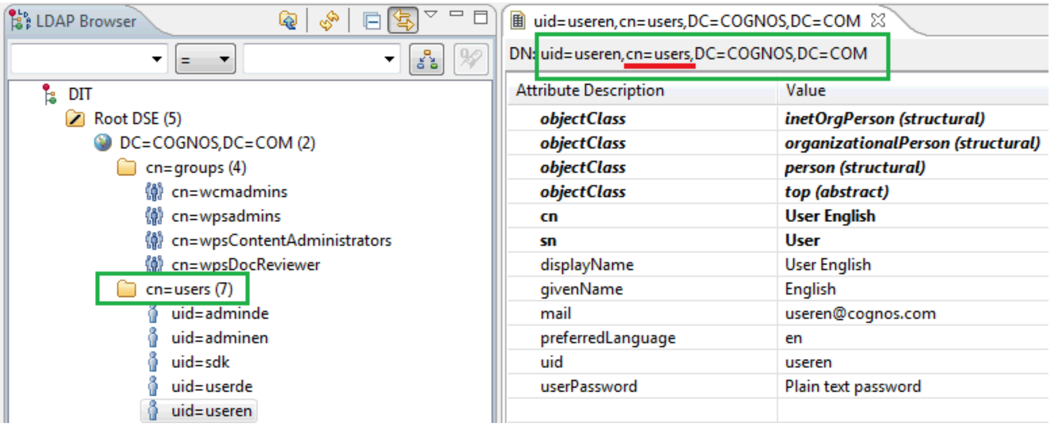
And as a result an error has been displayed:



'Search' has encountered a problem.

Error while performing search
- [LDAP: error code 32 - No Such Object]

No such object, user can not be found, so something is wrong. In this case the fault is in base distinguished name, which is *ou=users,dc=cognos,dc=com* while is suppose to be *cn=users,dc=cognos,dc=com*

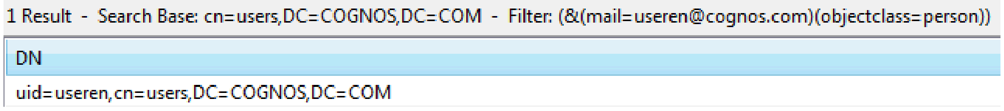


DIT
- Root DSE (5)
  - DC=COGNOS,DC=COM (2)
    - cn=groups (4)
      - cn=wcmadmins
      - cn=wpsadmins
      - cn=wpsContentAdministrators
      - cn=wpsDocReviewer
    - cn=users (7)
      - uid=adminde
      - uid=adminen
      - uid=sdk
      - uid=userde
      - uid=useren

DN: uid=useren,cn=users,DC=COGNOS,DC=COM

Attribute Description	Value
objectClass	***inetOrgPerson (structural)***
objectClass	***organizationalPerson (structural)***
objectClass	***person (structural)***
objectClass	***top (abstract)***
cn	**User English**
sn	**User**
displayName	User English
givenName	English
mail	useren@cognos.com
preferredLanguage	en
uid	useren
userPassword	Plain text password

After correction search result is correct, LDAP namespace is properly configured



1 Result - Search Base: cn=users,DC=COGNOS,DC=COM - Filter: (&(mail=useren@cognos.com)(objectclass=person))

DN
uid=useren,cn=users,DC=COGNOS,DC=COM

There is a separate technote which contains more information on how to use macros and special characters in both *User lookup* and *External identity mapping* : http://www-01.ibm.com/support/docview.wss?uid=swg21649894

Use external identity

If this property is set to true, the user is authenticated by an external source and the user's identity is provided from the external source. For example, if Single Sign On is configured, then Web server sets the REMOTE_USER as user's identity. If you set this property to true, ensure that you set the "External Identity Mapping" property.

External identity mapping

Value of this property should be built very similar way as *User lookup* string(see above), but instead of *${userID}* string should contain *${environment("ENVIRONMENT_VARIABLE_NAME")*, for example *${environment("REMOTE_USER")}*.

In some cases it may be required to modify *${environment("ENVIRONMENT_VARIABLE_NAME")* before it will be used in search string.

For example when *REMOTE_USER* is returned as *DOMAIN\user* and we would like to use *user* in search string then replace() function can be used, for example: *${replace(${environment("REMOTE_USER")},"DOMAIN\\","")}*

There is a separate technote which contains more information on how to use macros and special characters in both *User lookup* and *External identity mapping* : http://www-01.ibm.com/support/docview.wss?uid=swg21649894

Bind user DN and password

Specifies the credentials used for binding to the LDAP server when performing a search using the user lookup property, or when performing all operations using the external identity mapping.

Size limit
Specifies the maximum number of responses for a search request. When the size limit is reached the directory server stops searching. The default value -1 indicates that Cognos does not limit it and the value on the LDAP server will be used.

Time out in seconds
Specifies the number of seconds permitted to perform a search request. If the duration is exceeded, the search is timed out. The default value -1 indicates that Cognos does not limit it and the value on the LDAP server will be used.

UseBindCredentialsForSearch
This property affects users who don't use the external identity mapping. If this property is set to true, the *Bind user DN and password* will be used to perform a search in the LDAP directory server. If this flag is false or bind credentials are not presented, the authenticated user credentials will be used for searching

Allow Empty Password
Specifies whether empty passwords are allowed for user authentication

Unique identifier
Specifies the value used to uniquely identify objects stored in the LDAP directory server.

If an attribute is used, it must
- exist for all objects, such as users, groups, folders
- and must be unique.

A good example of such attribute is Active Directory *objectGUID* attribute.

If the *'dn'* is used, then full distinguished name is used e.g . *uid=useren,cn=users,dc=cognos,dc=com* .The advantage of using *dn* is that it provides unique identifier without any extra step. The disadvantage of this is that if object like user, group od folder is moved to different location in LDAP tree, then 'dn' will not match and Cognos policies may be affected**.**

Example 1:

- Unique identifier = dn (so Cognos will use full dn path as an unique identifier: *uid=useren,cn=users,dc=cognos,dc=com*)
- User account has been moved from *cn=users,dc=cognos,dc=com* to *cn=contractors,dc=cognos,dc=com*

In this case Cognos will treat user's account as a new one since *dn* path has changed.

Example 2:

- Unique identifier = *objectGUID* (so Cognos will use *objectGUID* as an unique identifier)
- User account has been moved from *cn=contractors,dc=cognos,dc=com* to *cn=users,dc=cognos,dc=com*

In this case Cognos will treat user's account as a the same account even if *dn* path has changed.

Data encoding

Specifies the encoding of the data stored in the LDAP directory server.
For example, use windows-1252, iso-8859-1, iso-8859-15, Shift_JIS, utf-16, or utf-8.

SSL Certificate Database

Specifies the location of the certificate database used by the directory server for SSL connections.
There is a separate technote on this:
http://www-01.ibm.com/support/docview.wss?uid=swg21344083

Folder mappings

This section contains configuration settings used by Cognos to identify and properly describe LDAP objects as folders.

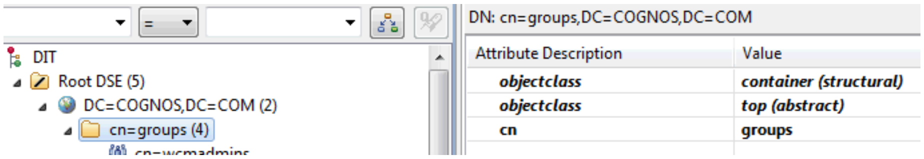
Object Class

Specifies the LDAP object class value used to identify LDAP object as a folder.
This object class:
- must exists for all LDAP folders
- must not exists for other objects like users and groups

If this value match, then Cognos will treat LDAP object as a folder. If this value is not properly configured Cognos will not display any folder in namespace.

If there is more types of objects to be treated as a folder, list of object classes can be used.
For example, if LDAP containers, organizations and organizational units needs to be treated by Cognos as folder, then all classes can be added as a list separated by coma. Then:
objectclass=urganizationaUnit,organization,container

In our example we will map LDAP containers as folders, thus object class: container, what can be verified in a LDAP browser:



DN: cn=groups,DC=COGNOS,DC=COM

Attribute Description	Value
objectclass	container (structural)
objectclass	top (abstract)
cn	groups

Description

Specifies the LDAP attribute used as a description of certain folder in Cognos. By default it is *description* attribute.

Name

Specifies the LDAP attribute used as a name of certain folder in Cognos. In our example it is *cn* attribute (see image above).

Group mappings

This section contains configuration settings used by Cognos to identify and properly describe LDAP objects as groups.

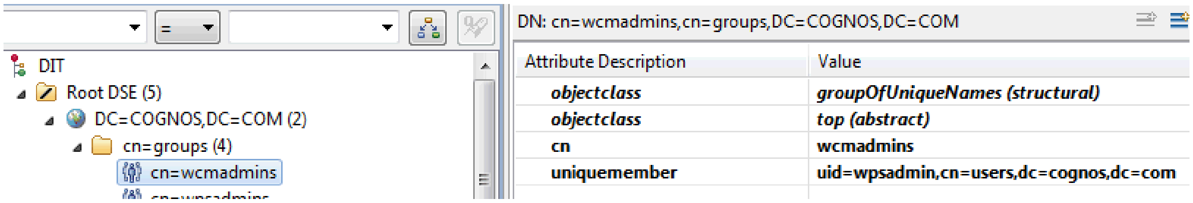
Object Class

Specifies the LDAP object class value used to identify LDAP object as a group.
This object class:
- must exists for all LDAP groups
- must not exists for other objects like users and folders

If this value match, then Cognos will treat LDAP object as a group. If this value is not properly configured Cognos will not display any group in namespace.

If there is more types of objects to be treated as a group, list of object classes can be used. For example, if LDAP group, groupOfUniqueNames and posixgroup needs to be treated by Cognos as a group, then all classes can be added as a list separated by coma. Then:
objectclass=group,posixGroup,groupOfUniqueNames

In our example we will use object class: *groupOfUniqueNames*, what can be verified in a LDAP browser as below:



DN: cn=wcmadmins,cn=groups,DC=COGNOS,DC=COM

Attribute Description	Value
objectclass	*groupOfUniqueNames (structural)*
objectclass	*top (abstract)*
cn	wcmadmins
uniquemember	uid=wpsadmin,cn=users,dc=cognos,dc=com

Description

Specifies the LDAP attribute used as a description of certain group in Cognos. By default it is *description* attribute.

Member

Specifies the LDAP attribute to identify a member of a group. In our example *uniquemember* attribute is used to identify members, see image above. If this value is not properly configured, Cognos will may display groups but not be able to display group members.

Name

Specifies the LDAP attribute used as a name of certain folder in Cognos. In our example it is *cn* attribute, see image above.

User mappings

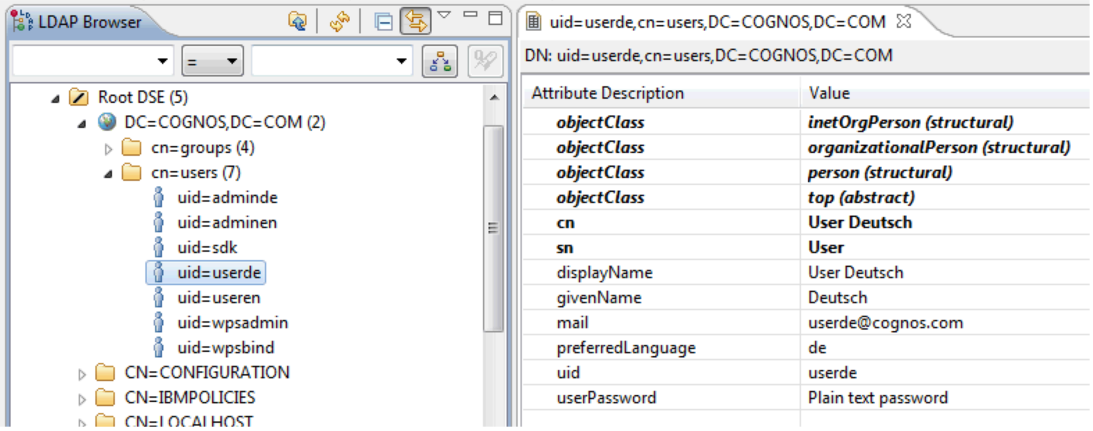
This section contains configuration settings used by Cognos to identify and properly describe LDAP object as an user.

Account Object Class

Specifies the LDAP object class value used to identify LDAP object as an user.
This object class:
- must exists for all LDAP users
- must not exists for other objects like groups and folders

If this value match, then Cognos will treat LDAP object as a group. If this value is not properly configured Cognos will not be able to display any user in namespace.

In our example we will use object class: *inetOrgPerson*, what can be verified in a LDAP browser:



DN: uid=userde,cn=users,DC=COGNOS,DC=COM

Attribute Description	Value
objectClass	***inetOrgPerson (structural)***
objectClass	***organizationalPerson (structural)***
objectClass	***person (structural)***
objectClass	***top (abstract)***
cn	**User Deutsch**
sn	**User**
displayName	User Deutsch
givenName	Deutsch
mail	userde@cognos.com
preferredLanguage	de
uid	userde
userPassword	Plain text password

Using a LDAP browser it can be found that *objectClass=inetOrgPerson* can be used. But also *objectClass=person* or *objectClass*=organizationalPerson can be correct if only this class exists for all users and no other LDAP objects.

Other values as: Business phone, Content Locale, Description, Email, Fax/Phone, Given name, Home phone, Mobile phone, Name, Pager phone, Password, Postal Address, Product locale, Surname, User name

Specifies additional details for user account. Map them to best matching attributes from LDAP. Use of a LDAP viewer is very helpful to figure it out.

Not all of them must exist in LDAP, in our example we may map below Cognos fields to LDAP attributes:

- Email to *mail*
- Given name to *givenname*
- Name to *cn*
- Password to *userPassword*
- Product locale to *preferredLanguage*
- Surname to *sn*
- User name to *uid*

The rest of Cognos fields does not have any equivalent in LDAP, but this is not required to map all of them.