

Querying Active Directory Through SQL Server

Integrating your application with Active Directory (or another LDAP based directory) is a common requirement in many business applications. Almost every language has a way to query LDAP but little known is the approach of integrating SQL Server with your LDAP controller. This quick article will guide you through setting up and using LDAP queries through SQL Server 2000 with Active Directory as the LDAP controller.

First, you need to create a SQL Server linked server. Set it up through SQL Server Enterprise Manager:

1. Open SQL Server Enterprise Manager
2. Go to the database server to which you will be adding the linked server
3. Expand "Security"
4. Right-click on "Linked Servers" and click on "New Linked Server..."
5. Fill in the following:
 1. Under the General tab:
 1. Linked Server: **adsi** (or whatever you want to call it)
 2. Server type: select **Other Data Source**
 3. Provider name: select **OLE DB Provider for Microsoft Directory Services**
 4. Under Provider Options: check that **Allow InProcess** is checked
 5. Leave the rest of the fields blank
 2. Under the Security tab:
 1. Local Login: **sqlServerUser**
 2. Remote User: ntaccount@domain.com (such as bkostadinov@ica.com or ica.com\bkostadinov)
 3. Remote Password: **userPassword** for the above domain account

Through Query Analyzer:

1. Change the "AllowInProcess" registry key under **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Providers\ADSDSOObject** to "**dword:00000001**"
 1. Edit the registry manually or put the following in a .reg file and execute it:

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Providers\ADSDSOObject]

"AllowInProcess"=dword:00000001
```
2. Open Query Analyzer (or your choice of query tool)
3. Connect to the server to which you will be adding the linked server
4. Change the provided values and execute the following code:

[view plaincopy to clipboardprint?](#)

```
1. -- Change 'adsi' to the desired name of the linked server
2. exec sp_addlinkedserver
3. 'adsi',
4. ",
5. 'ADSDSOObject',
6. "
7. go
8.
9. -- Change 'adsi' to the desired name of the linked server
10. -- Change 'sqlUser' to the username of local sql server user
11. -- Change 'domainName\userName' to a domain account
12. -- (the format can be 'domainName\userName' or 'userName@domainName')
13. -- Change 'domainUserPassword' to the password of the domain account
14. exec sp_addlinkedsrvlogin
15. 'adsi',
16. false,
17. 'sqlServerUser',
18. 'domainName\userName',
19. 'domainUserPassword'
20. go
```

Or you can just fill in the values on top of the following script and run that:

[view plaincopy to clipboardprint?](#)

```
1. declare @linkedServerSql nvarchar(4000),
2.   @linkedServerName varchar(100),
3.   @localSqlUsername varchar(100),
4.   @domainUsername varchar(100),
5.   @domainUserPassword varchar(100)
6.
7. -- Set the local sql server user
8. set @localSqlUsername = 'sqlUser'
9. -- format can be 'domainName\userName' or 'userName@domainName'
10. set @domainUsername = 'domainName\userName'
11. set @domainUserPassword = 'domainUserPassword'
12.
13. set @linkedServerName = 'adsi'
14.
15. set @linkedServerSql = '
16.   exec sp_addlinkedserver
17.   ''' + @linkedServerName + ',
18.   ',
19.   "ADSDSOObject",
20.   '''
21.
22. exec sp_addlinkedsrvlogin
23.   ''' + @linkedServerName + ',
24.   false,
25.   ''' + @localSqlUsername + ',
26.   ''' + @domainUsername + ',
27.   ''' + @domainUserPassword + '''
28.
29. exec sp_executesql @linkedServerSql
```

Run a query to verify that the linked server works. The query below will give you all the users in the **dc=ica,dc=com** (change that to match your own domain):

[view plaincopy to clipboardprint?](#)

```
1. select *
2. from openquery(adsi, '
3. select givenName,
4.     sn,
5.     sAMAccountName,
6.     displayName,
7.     mail,
8.     telephoneNumber,
9.     mobile,
10.    physicalDeliveryOfficeName,
11.    department,
12.    division
13. from   "LDAP://dc=ica,dc=com"
14. where  objectCategory = "Person"
15.       and
16.       objectClass = "user"
17.')
```

Below, is an alternate syntax which you can use to apply ldap filters with almost universal syntax. The following will get all the users in LDAP but limit the result set to those users who's "given" & "sn" names are not empty. It will also apply a filter to the "division" attribute and exclude any records that match "system" and "generic".

[view plaincopy to clipboardprint?](#)

```
1. declare @ldapFilter nvarchar(1000), @ldapSQL nvarchar(4000)
2.
3. --
   Set the filter to exlude objects that have a division of "System" and "Generic"

4. set @ldapFilter = '(!division=System*)(!division=Generic)'
5.
6. -- Create an ldap query to get all users under dc=ica,dc=com
7. set @ldapSQL = '
8. select  givenName as firstName,
9.         sn as lastName,
10.        displayName,
11.        lower(sAMAccountName) as accountName,
12.        telephoneNumber as phoneNumber,
13.        mobile as cellPhoneNumber,
14.        mail as emailAddress,
15.        department,
16.        physicalDeliveryOfficeName as siteName
17. from    openquery(ads, "<LDAP://dc=ica,dc=com>
18.        (&(objectCategory=Person)(objectClass=user))+ @ldapFilter + ');
19.        givenName,
20.        sn,
21.        sAMAccountName,
22.        displayName,
23.        mail,
24.        telephoneNumber,
25.        mobile,
26.        physicalDeliveryOfficeName,
27.        department,
28.        division;
29.        subtree")
30. where   givenName is not null
31.        and
32.        sn is not null'
33.
34. exec sp_executesql @ldapSQL
```

Some things to note:

"LDAP" is case sensitive, if you try using "ldap", the query will throw an error.

Querying the Active Directory server will work fine from Query Analyzer even if you do not run **sp_addlinkedserverlogin**. However, if you try to execute the query from the web (with something like ColdFusion, you will get an error similar to:

*[Macromedia][SQLServer JDBC Driver][SQLServer]OLE DB provider
'ADSDSOObject' reported an error. The provider indicates that the user did not have
the permission to perform the operation.*