

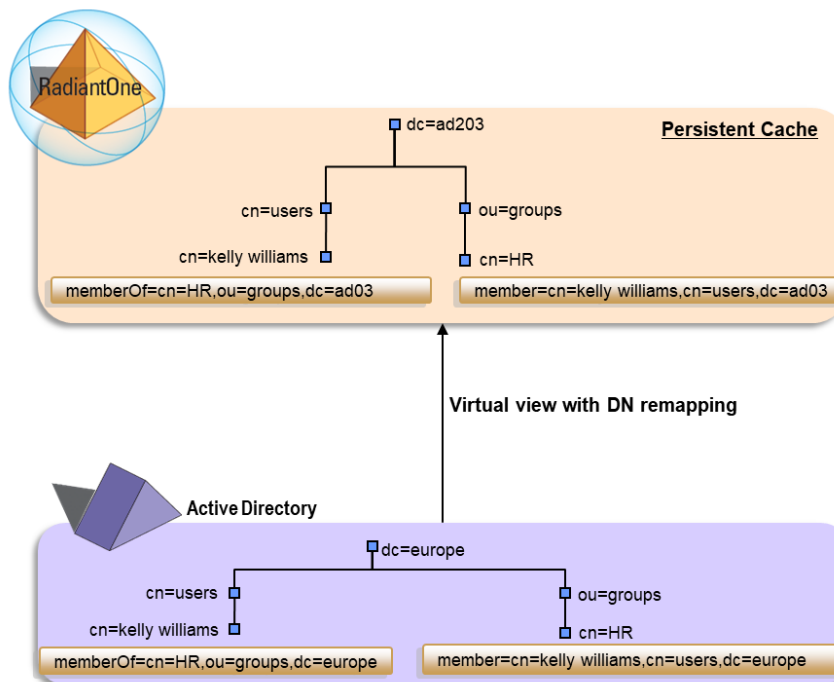
Calculating MemberOf for Cached Users from Active Directory

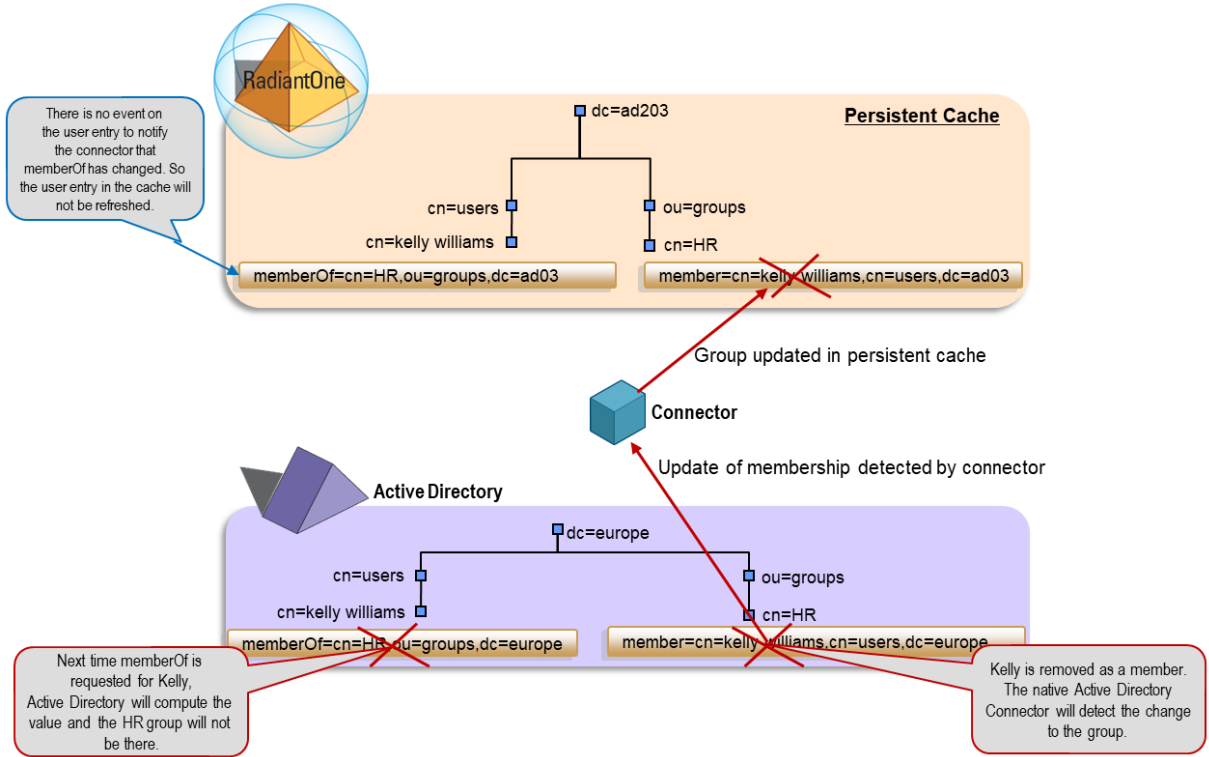
Overview	2
Configuration	4
Create a Data Source for Active Directory.....	4
Create a Virtual View for Active Directory.....	4
Configure a Persistent Cache with Automated Refresh.....	6
Configure Special Attributes Handling for isMemberOf	8
Start the Connectors for the Real Time Automated Cache Refresh	11
Testing the Calculation of MemberOf.....	12
Support for Nested Groups - LDAP_MATCHING_RULE_IN_CHAIN.....	13
Support for Flattening Nested Groups.....	14

Overview

Depending on the number of groups and size of group entries in an LDAP directory, it may be more efficient for client applications to search the user entry for which groups the user is a member of. In Active Directory, the memberOf attribute in the user entries is computed dynamically by the server and contains the DN's for all groups the user is currently a member of.

When you define a persistent cache at the level of RadiantOne FID, for a virtual view from Active Directory, the user entries will have their memberOf attribute cached as well. However, since the memberOf attribute is dynamically computed by Active Directory, the RadiantOne native connectors cannot detect changes on the user entry if their group membership changes. The group entries that are cached from Active Directory will be refreshed in the persistent cache because the member attribute will be detected as updated by the native connector. However, the corresponding user entry in the persistent cache will not have their memberOf attribute updated properly. This is depicted in the following two diagrams.





IMPORTANT NOTE – if you are using the snapshot connector type as opposed to the native AD connector type, then both the group entry and the user entry will be properly updated in the persistent cache because snapshots read/compare the entire entry and will detect a changed value in memberOf.

If you are not able to use the snapshot connector type for persistent cache refresh, then you should have RadiantOne FID dynamically compute the membership attribute for the users instead. This document describes the configuration for this use case.


Configuration

The integration between RadiantOne and Active Directory is detailed in this guide and is broken into the following high-level configuration steps:

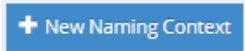
1. Create a data source/establish a connection to Active Directory.
2. Create a proxy view for the groups and users in Active Directory, indicating to hide the memberOf attribute returned.
3. Configure a persistent cache with automated refresh for the proxy view.
4. Configure a special attribute handling and index for isMemberOf.
5. Start the Persistent Cache Refresh components.

Note – These steps have been validated with RadiantOne v7.2.24 and Active Directory 2016.

Create a Data Source for Active Directory

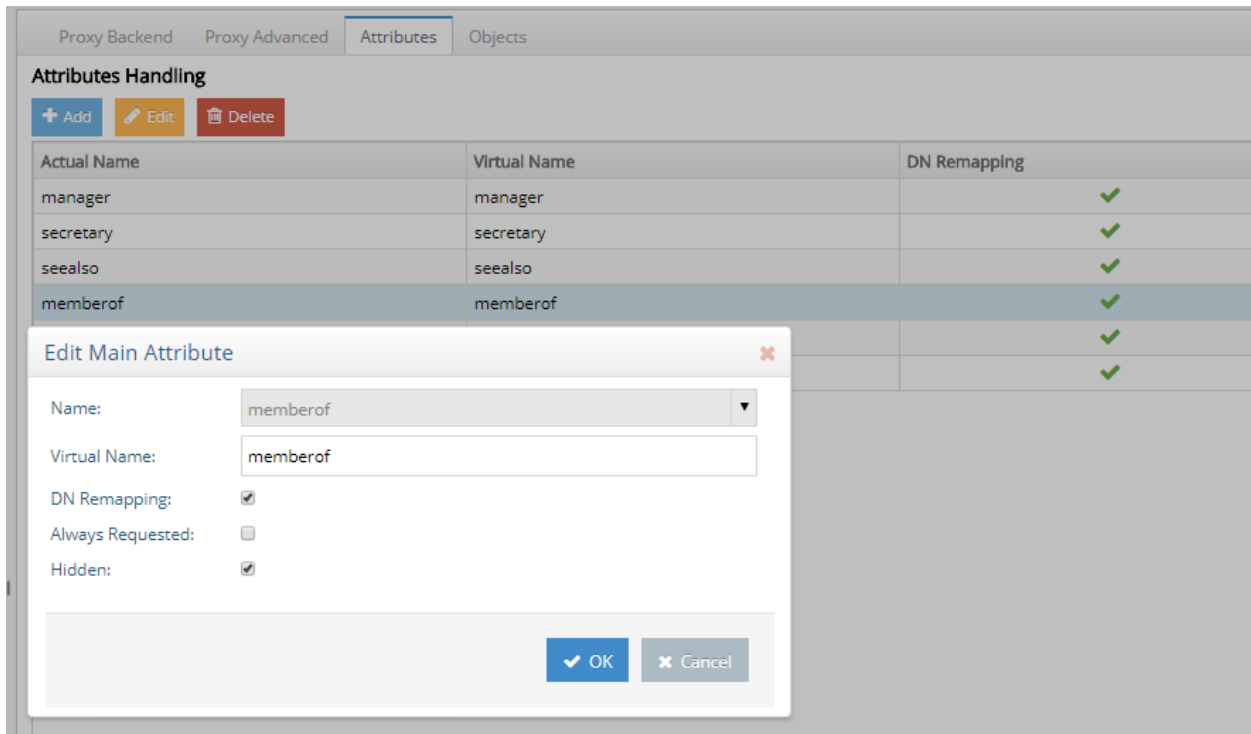
1. From the Main Control Panel -> Settings Tab -> Server Backend section, select LDAP Data Sources.
2. On the right side, click ADD.
3. Enter the connection details for Active Directory. The Base DN should be a container that encompasses all user and group accounts in your Active Directory. In the context of this document, the data source name is seradiant.
4. Click  .


Create a Virtual View for Active Directory

1. From the Main Control Panel -> Directory Namespace Tab, click  .
2. Enter a naming context (e.g. o=seradiant) and choose LDAP Backend for the type.
3. Click Next.
4. Choose the data source you created in the previous section from the Data Source drop-down list and click OK.
5. On the Directory Namespace tab, select this new naming context and click on the

Attributes tab.

6. Click on memberOf and click EDIT.
7. Check the Hidden option for this attribute and click OK.

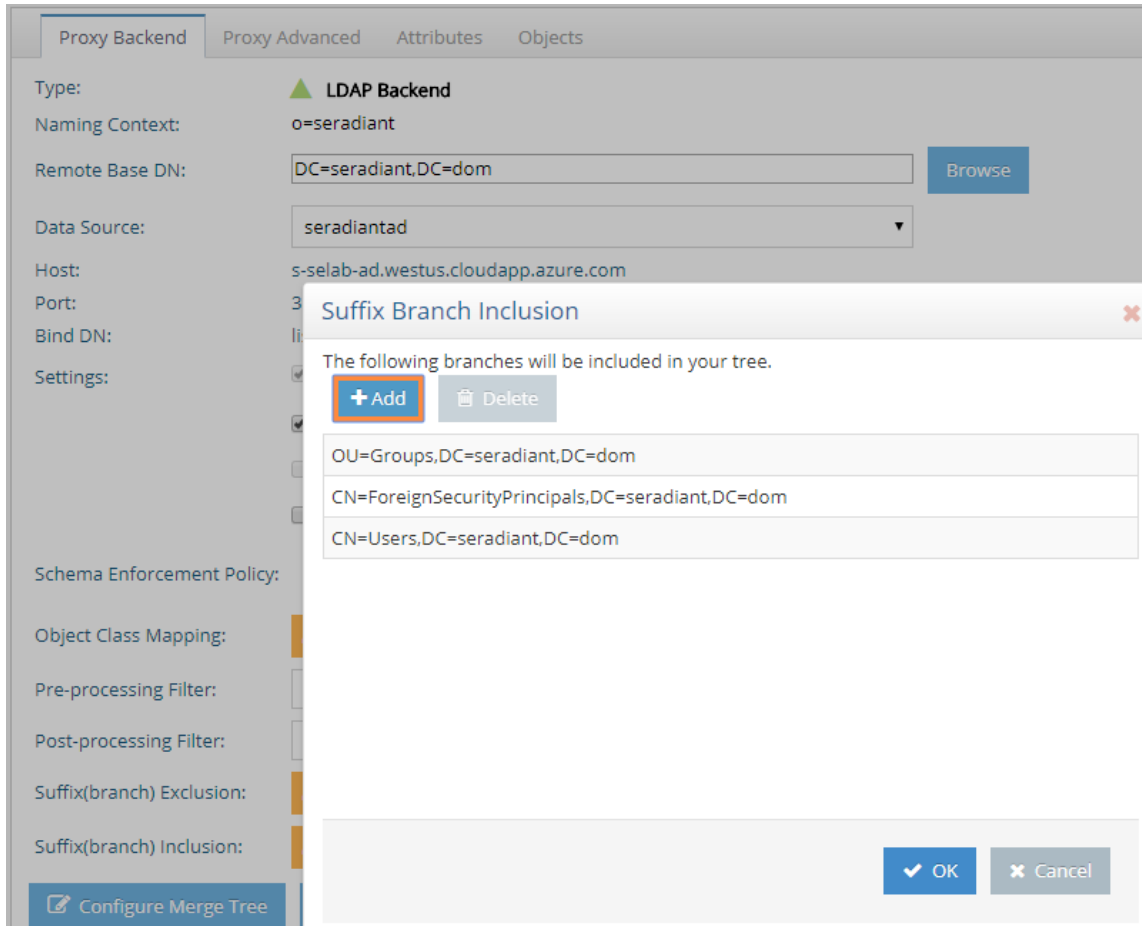


8. Click  .
9. Click Yes to apply the changes to the server.

IMPORTANT NOTE – if you create the view using a “virtual tree” type of backend (or with Context Builder), instead of a proxy approach like described here, you will need to define computed attributes for the member and memberOf attributes to remap the suffix to match the naming used in the RadiantOne virtual namespace. See the Radiant Logic Knowledge Base (<http://www.radiantlogic.com/support/knowledge-database/>) for steps on creating these computed attributes.

10. (Optional) To limit the branches virtualized from Active Directory, you can go to the Proxy Backend tab and use the Suffix (branch) Exclusion or Suffix (branch) Inclusion settings. In the example below, the virtual view only includes the OU=Groups,DC=seradiant,DC=dom,

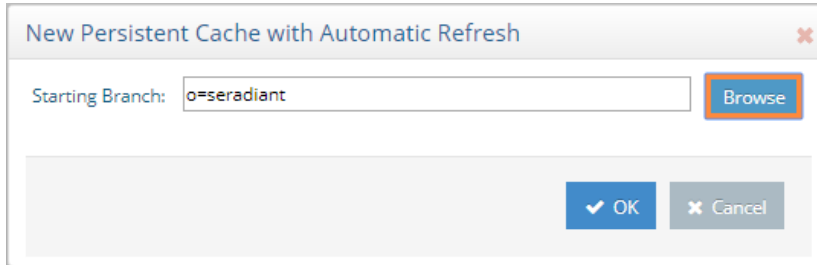
CN=ForeignSecurityPrincipals,DC=seradiant,DC=dom, and
 CN=users,DC=seradiant,DC=dom branches.



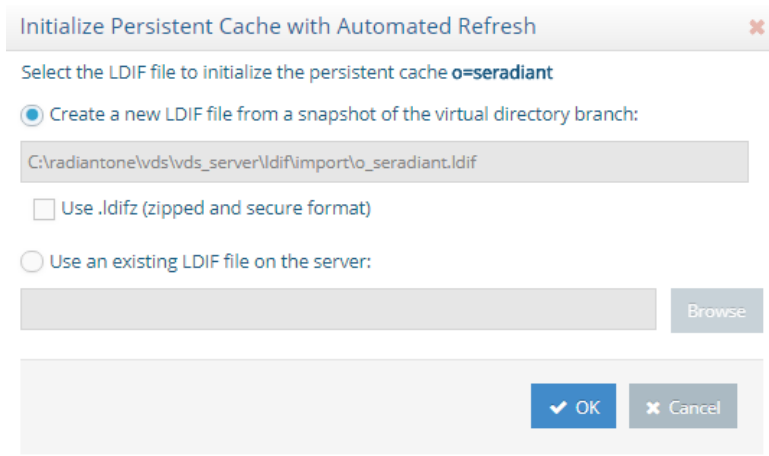
Configure a Persistent Cache with Automated Refresh

Make sure GlassFish is running prior to continuing. You can check the status of GlassFish from the Main Control Panel -> Syn Monitoring tab. For assistance with starting GlassFish, see the RadiantOne Deployment and Tuning Guide.

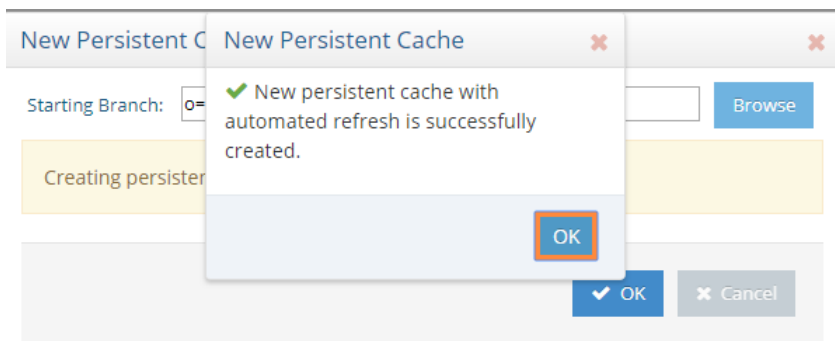
1. In the Cache section at the bottom of the Directory Namespace tab, select .
2. On the right side, select **Persistent Cache with Automated Refresh** and click .
3. Click Browse to select the naming context you created in the previous section.



4. Click OK.
5. Click OK to close the confirmation of the new persistent cache configuration.
6. Select the cached branch below and on the right, click .
7. Select the option to “Create a new LDIF file...” and click OK.



8. Wait until the cache initialization has finished and click OK.

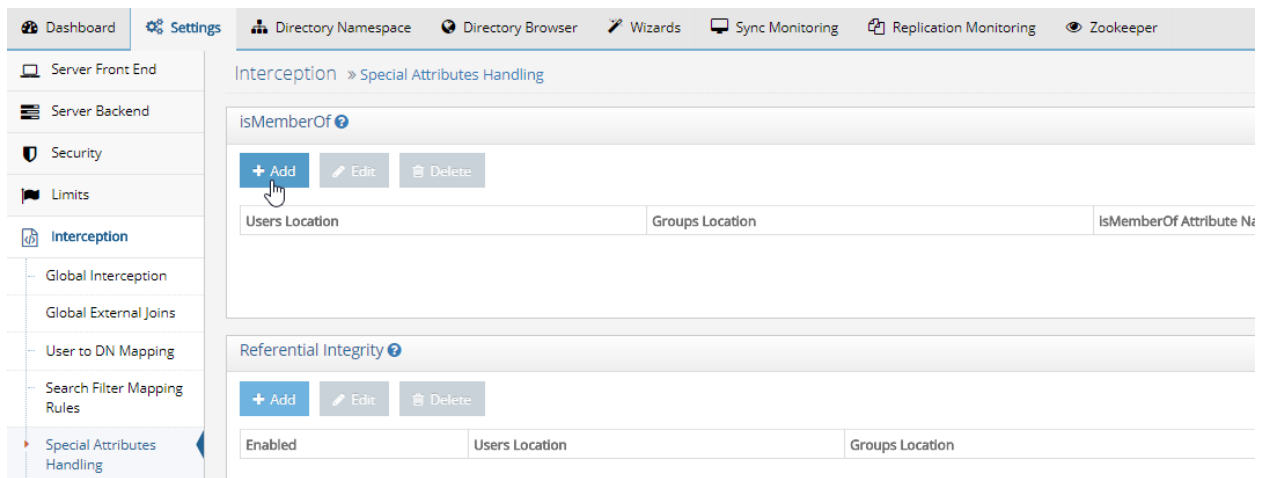


Configure Special Attributes Handling for isMemberOf

The attribute storing the list of groups the user is a member of is computed by RadiantOne FID and returned as `isMemberOf` by default. You can configure this to be any attribute name you need (e.g. `memberOf`).

To calculate a list of groups a user is a member of, configure rules from the RadiantOne Main Control Panel.

1. From the Main Control Panel -> Settings Tab -> Interception section, click Special Attributes Handling. Locate the `isMemberOf` section and click on the Add button.



2. Click Choose next to Users Location. The Choose your base DN window is displayed.
3. Select a base DN for your users and click OK. (e.g. `cn=Users,o=seradiant`)
4. Under Groups Location, click Add. The Choose your base DN window is displayed.
5. Select a base DN for the groups location and click OK.
6. For the `isMemberOf` Attribute name, enter the attribute name you want the membership info to be returned in (e.g. `memberOf`).


Add Mapping

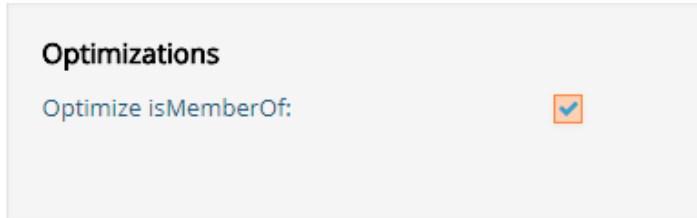
Users Location

Groups Location

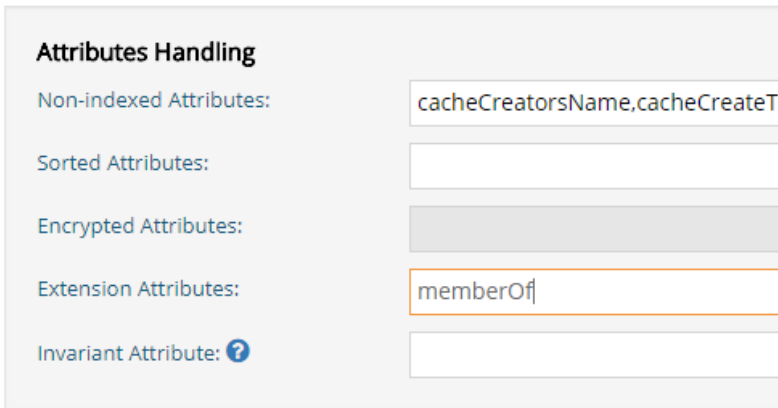
isMemberOf Attribute Name

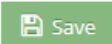
Static Filter

7. Click OK to close the Mapping window. You are returned to the Special Attributes Handling page.
8. Click  in the upper right corner of the Special Attributes Handling page.
9. Go back to the Main Control Panel -> Directory Namespace Tab -> Cache section and select the persistent cache branch.
10. On the Properties tab on the right, check the option to Optimize isMemberOf.



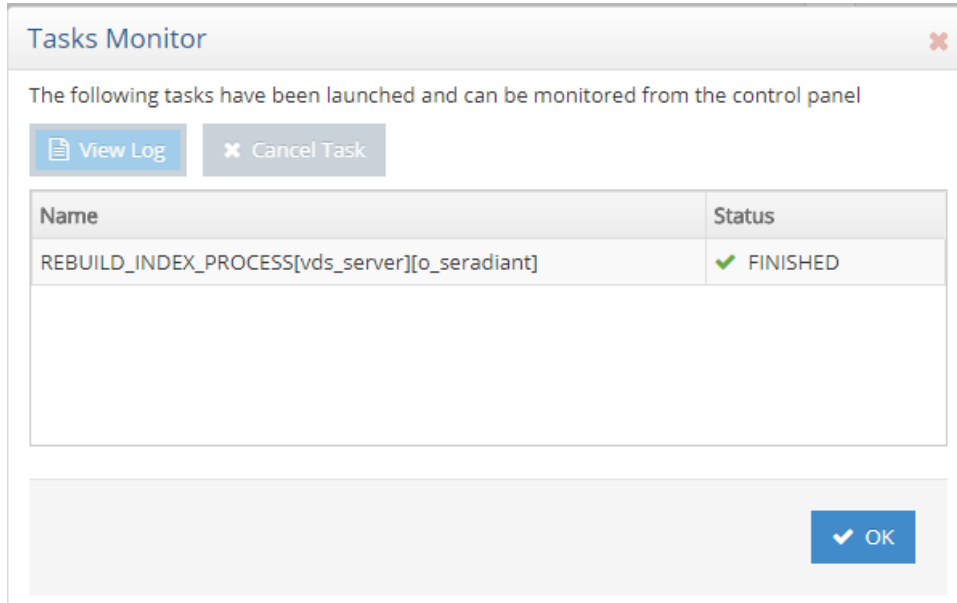
11. In the Attribute Handling section, enter the attribute name that contains the membership information that you configured in step 6 above.



12.  Save and then Yes to apply the changes to the server.


13. Click .

14. Click OK once the task has finished.



Start the Connectors for the Real Time Automated Cache Refresh

The last step is to start the connectors for the persistent cache refresh. This keeps the RadiantOne persistent cache refreshed when changes to users and groups occur in the Active Directory backend.


1. In the Main Control Panel, Directory Namespace tab, navigate below the cache node and select the persistent cache branch.
2. On the right side, click on the Connectors tab.
3. (Optional) To optimize the persistent cache refresh process, if you chose to use [Suffix \(branch\) Exclusion or Suffix \(branch\) Inclusion](#) in your proxy view, you can also add this condition for the Active Directory capture connector. Select the “Capture [AD Connector]” in the list and click  .
4. Locate the applicable “Excluded Branches” or “Included Branches” setting (matching whatever type of exclusion/inclusion you configured for your proxy view). Enter the list of branches in the applicable setting using a <space>##<space> to separate the branches. Below is an example that matches the configuration described in [Suffix \(branch\) Exclusion or Suffix \(branch\) Inclusion](#) used in this guide.

Capture Connector Settings	
Connector Type:	96
Publisher Topic Name:	CF_O_SERADIANT_so_o_seradiant_generic
Synchronization Object Name:	so_o_seradiant
Topology Name:	CF_O_SERADIANT
Detect New Changes Only[true/false]:	<input type="text" value="true"/>
Excluded attributes:	<input type="text"/>
Excluded branches:	<input type="text"/>
Included branches:	<input type="text" value="OU=Groups,DC=seradiant,DC=dom ## CN=ForeignSecurityPrincipals,DC=seradiant,DC=dor"/>
LDAP Filter:	<input type="text"/>

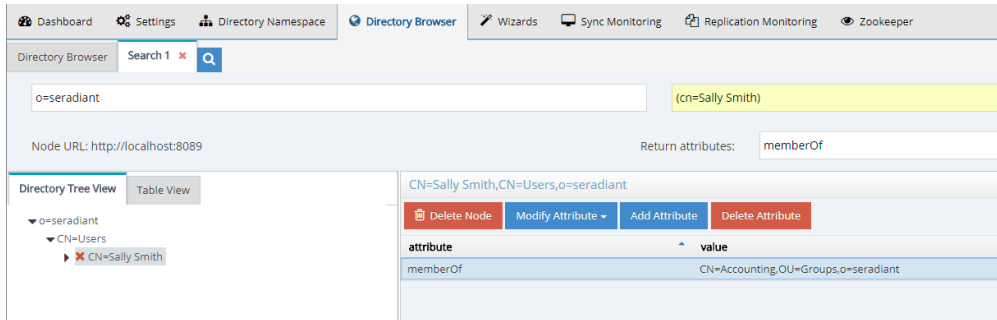
5. Click OK to save the capture connector settings.
6. Click the Start All button to start the persistent cache refresh process.

Testing the Calculation of MemberOf

The RadiantOne Main Control Panel -> Directory Browser tab can be used as an LDAP client to test the request of the memberOf attribute.

1. From the Main Control Panel, go to the Directory Browser tab.
2. Select your root naming context (e.g. o=seradiant) and click  .
3. In the search window, click Show advanced search .
4. Replace (objectclass=*) with a filter for a specific user (e.g. cn=Sally Smith).
5. Select Subtree to indicate the scope of the search.
6. Enter memberOf for the Return Attributes.
7. Click Q Search .
8. The search results are displayed below on the Directory Tree View tab in the lower left corner.
9. Click the user entry that is returned (e.g. CN=Sally Smith). The groups that Sally is

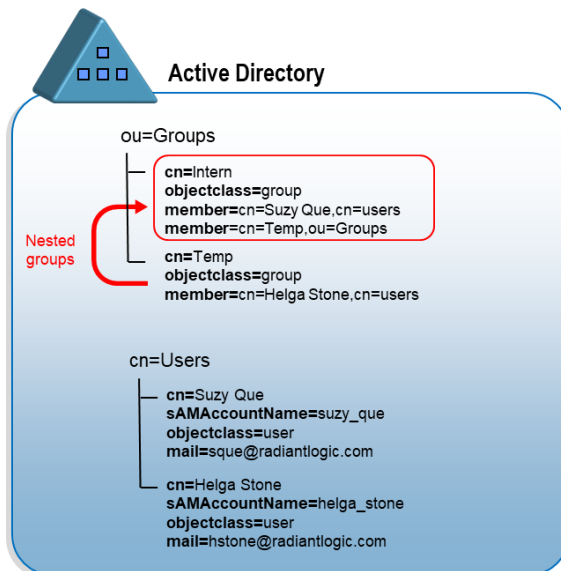
currently a member of (computed by RadiantOne FID), are displayed when you click on the entry. In the example shown below you can see that Sally is a member of CN=Accounting,OU=Groups,o=seradiant.



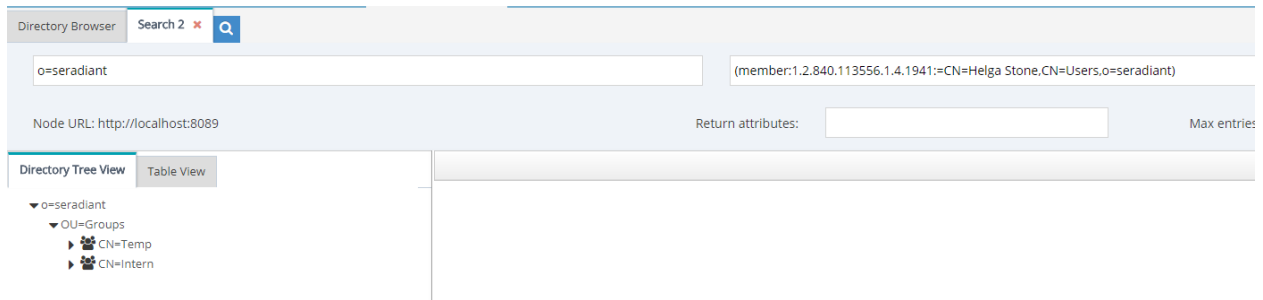
Support for Nested Groups - LDAP_MATCHING_RULE_IN_CHAIN

RadiantOne FID supports the LDAP_MATCHING_RULE_IN_CHAIN operator and allows clients to issue search filters using the 1.2.840.113556.1.4.1941 matching rule OID. This provides a method to look up the ancestry of an object and can be used in a search filter to retrieve all groups a user is a member of even when that group is nested (and is a member of another group).

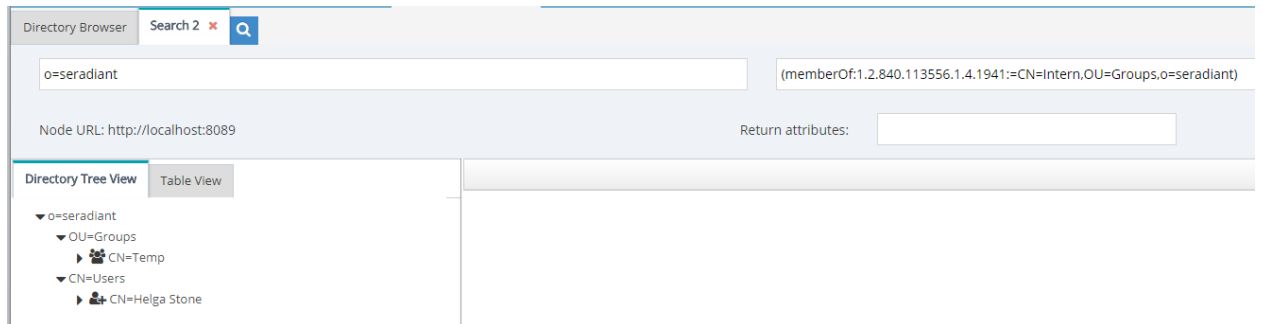
Active Directory supports nested groups, and in the example shown in the diagram below, the Temp group is a member of the Intern group. A user named Helga Stone is a member of the Temp group, which means she is also indirectly a member of the Intern group.



Clients can issue searches containing filters like (member:1.2.840.113556.1.4.1941:=CN=Helga Stone,CN=Users,o=seradiant) to return all the groups that a user is a member of. An example is shown below where both the Temp and Intern groups are returned.

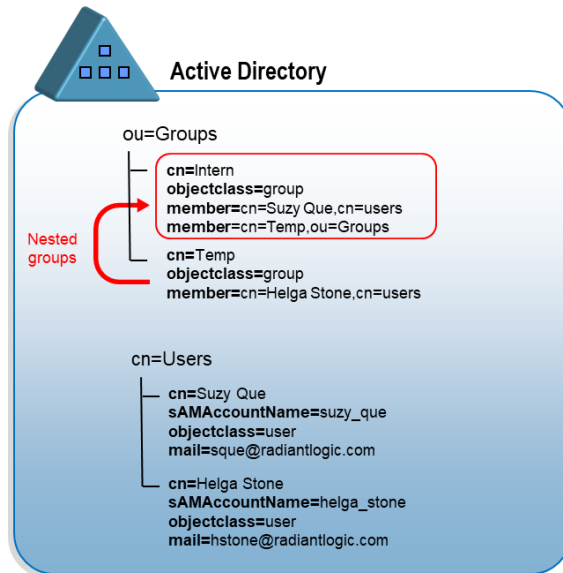


Clients can also use memberOf in the search filter like (memberOf:1.2.840.113556.1.4.1941:=CN=Intern,OU=Groups,o=seradiant) to return all users that are a member of a particular group. An example is shown below where both the Temp group and the user Helga Stone are returned as members of the Intern group.




Support for Flattening Nested Groups


Active Directory supports nested groups, in which case, groups can be members of other groups. In the example shown in the diagram below, the Temp group is a member of the Intern group. A user named Helga Stone is a member of the Temp group, which means she is also indirectly a member of the Intern group.




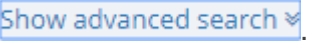
Some LDAP clients can't work with nested groups and don't support using the LDAP_MATCHING_RULE_IN_CHAIN filter described in the [previous section](#). In this case, RadiantOne can be configured to automatically flatten nested groups so the client can retrieve this information un-nested.

To enable support for nested groups in RadiantOne, check the option on the Main Control Panel -> Settings tab -> Security section -> Access Control sub-section and click .

Enable Nested Groups

Since the context of this guide uses a special index for storing group membership references, the index must be rebuilt after you enable support for nested groups. Go to the Main Control Panel -> Directory Namespace tab -> Cache node -> and select the persistent cache view from Active Directory. On the Properties tab on the right, click . Click OK after the process is finished.

To test the query for group membership, follow the steps below.

1. From the Main Control Panel, go to the Directory Browser tab.
2. Select your root naming context (e.g. o=seradiant) and click .
3. In the search window, click .

4. Replace (objectclass=*) with a filter for a specific user that is a member of a nested group (e.g. cn=Helga Stone).
5. Select to indicate the scope of the search.
6. Enter memberOf for the Return Attributes.
7. Click .
8. The search results are displayed below on the Directory Tree View tab in the lower left corner.
9. Click the user entry that is returned (e.g. CN= Helga Stone). All groups (including nested) that Helga is currently a member of (computed by RadiantOne FID), are displayed when you click on the entry. In the example shown below you can see that Helga is a member of CN=Temp,OU=Groups,o=seradiant and CN=Intern,OU=Groups,o=seradiant.

The screenshot shows the Directory Browser interface. At the top, there is a search bar with "Search 2" and a magnifying glass icon. Below the search bar, there are two input fields: "o=seradiant" and "(cn=Helga Stone)". The Node URL is "http://localhost:8089" and the Return attributes are "memberOf".

The Directory Tree View shows a tree structure with "o=seradiant" expanded to "CN=Users", which is further expanded to "CN=Helga Stone".

The search results for "CN=Helga Stone,CN=Users,o=seradiant" are displayed in a table:

attribute	value
memberOf	CN=Temp,OU=Groups,o=seradiant
memberOf	CN=Intern,OU=Groups,o=seradiant