RadiantOne LDAP Replacement Guide

Chapter 1: Overview

The purpose of this handbook is to provide guidance for migrating from a legacy LDAP directory (OpenDJ, SunOne/ODSEE, IBM Tivoli Directory) into the RadiantOne Universal Directory. This document provides a general overview of the migration process. The steps provided in this handbook are not exhaustive and may vary depending on the use case.

Although the RadiantOne Universal Directory supports the standard LDAP v3 RFC and closely mimics the behavior of legacy LDAP directory implementations, slight variations are possible. These variations might impact client applications. The level of impact depends on how tightly-coupled the application is with the LDAP variations implemented by the legacy LDAP directory. In certain cases, the logic of the application might need to change. In some situations, where clients cannot change, RadiantOne's various customization techniques (e.g. interception scripts, computed attributes…etc.) can be used to mimic the legacy directory. If you encounter this situation, reach out to support@radiantlogic.com for assistance.

This gets you all of the components needed for your replacement task. Then, the outline below details the general migration strategy. Each item is further detailed in later chapters.

Chapter 2 - Inventory existing directory (schema, hierarchy, password policies…etc.)

Chapter 3 - Import data into RadiantOne Universal Directory

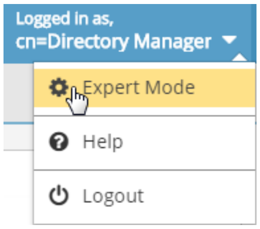
Chapter 4 - Configure RadiantOne server settings

Chapter 5 - Determine the application usage and cutover strategy

Chapter 6 - Decommission legacy directory

Expert Mode

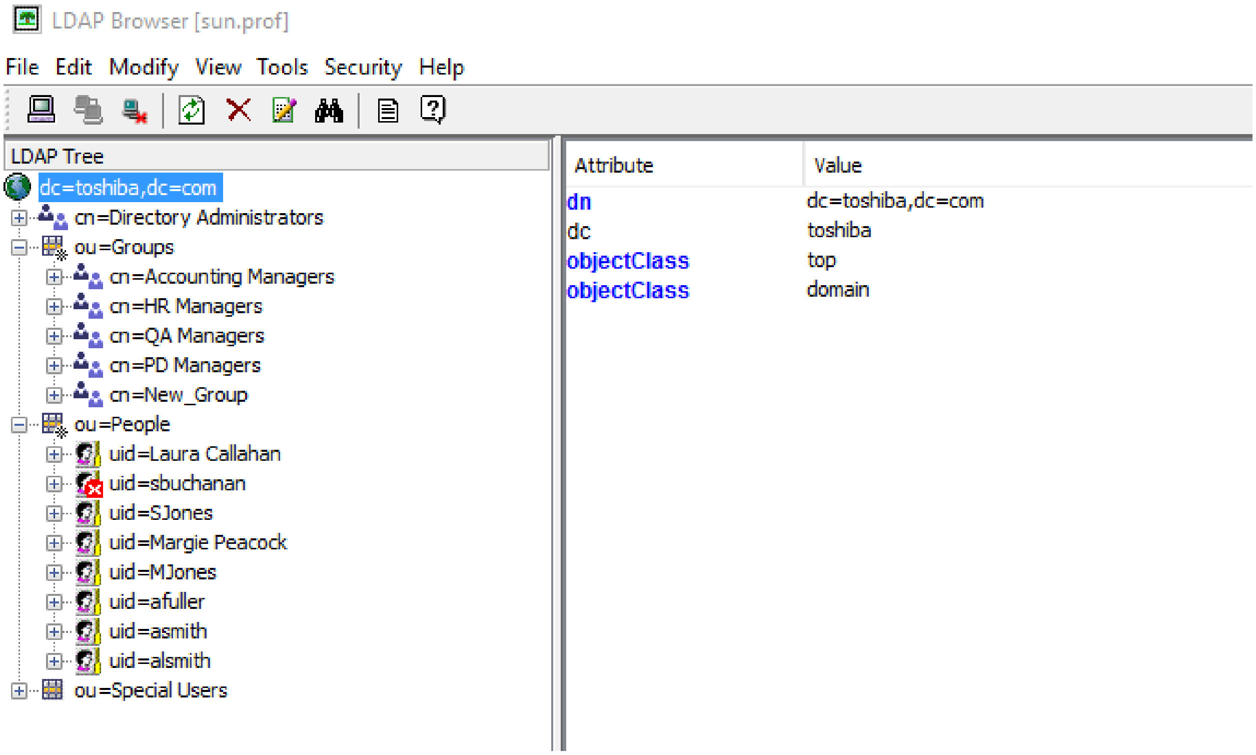
Some settings in the Main Control Panel are accessible only in Expert Mode. To switch to Expert Mode, click the Logged in as, (username) drop-down menu and select Expert Mode.



Note - The Main Control Panel saves the last mode (Expert or Standard) it was in when you log out and returns to this mode automatically when you log back in. The mode is saved on a per-role basis.

Chapter 2: Inventory Existing Directory

Taking inventory of the existing directory is mostly a manually process. Once you've acquired the basic credentials from the directory owner, you can access the directory from any LDAP client. RadiantOne includes the LDAP Browser client that can be used to connect to the directory. From here, you can get a glimpse of the existing hierarchy and export schema and branches to LDIF files. Below is an example of using the RadiantOne LDAP Browser.

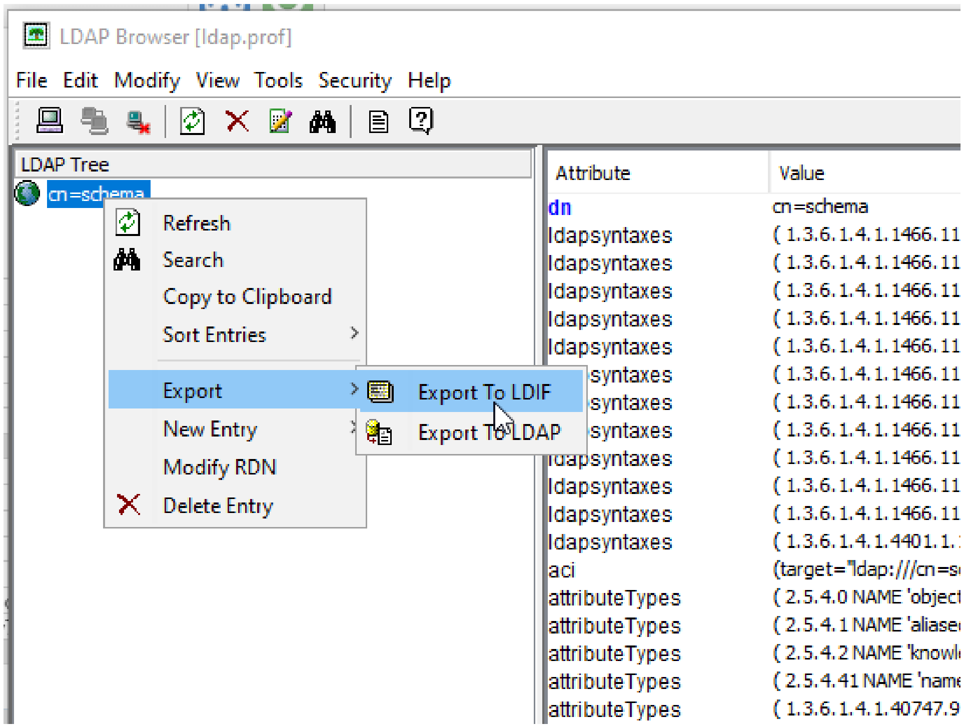


```
dc=toshiba,dc=com
  cn=Directory Administrators
  ou=Groups
    cn=Accounting Managers
    cn=HR Managers
    cn=QA Managers
    cn=PD Managers
    cn=New_Group
  ou=People
    uid=Laura Callahan
    uid=sbuchanan
    uid=SJones
    uid=Margie Peacock
    uid=MJones
    uid=afuller
    uid=asmith
    uid=alsmith
  ou=Special Users
```

Attribute	Value
dn	dc=toshiba,dc=com
dc	toshiba
objectClass	top
objectClass	domain

Schema

To get the schema information from the LDAP directory, use a base DN of cn=schema in LDAP Browser. Then, export the schema to LDIF from the right-click menu.



Attribute	Value
dn	cn=schema
ldapsyntaxes	(1.3.6.1.4.1.1466.11
ldapsyntaxes	(1.3.6.1.4.1.1466.11
ldapsyntaxes	(1.3.6.1.4.1.1466.11
ldapsyntaxes	(1.3.6.1.4.1.1466.11
ldapsyntaxes	(1.3.6.1.4.1.1466.11
syntaxes	(1.3.6.1.4.1.1466.11
syntaxes	(1.3.6.1.4.1.1466.11
syntaxes	(1.3.6.1.4.1.1466.11
ldapsyntaxes	(1.3.6.1.4.1.1466.11
ldapsyntaxes	(1.3.6.1.4.1.1466.11
ldapsyntaxes	(1.3.6.1.4.1.1466.11
ldapsyntaxes	(1.3.6.1.4.1.4401.1.
aci	(target="ldap:///cn=s
attributeTypes	(2.5.4.0 NAME 'object
attributeTypes	(2.5.4.1 NAME 'aliase
attributeTypes	(2.5.4.2 NAME 'knowl
attributeTypes	(2.5.4.41 NAME 'name
attributeTypes	(1.3.6.1.4.1.40747.9

LDAP Controls

Understanding the enabled LDAP controls (e.g. paged results, VLV/sort, persistent search, proxy authorization) is a manual process. Check the legacy directory server settings to determine which controls are enabled.

Password Policies

Understanding the password policies defined in the legacy directory is a manual process. You must work with the directory owner/administrator to understand how password policies are enforced. Some questions to ask might be:

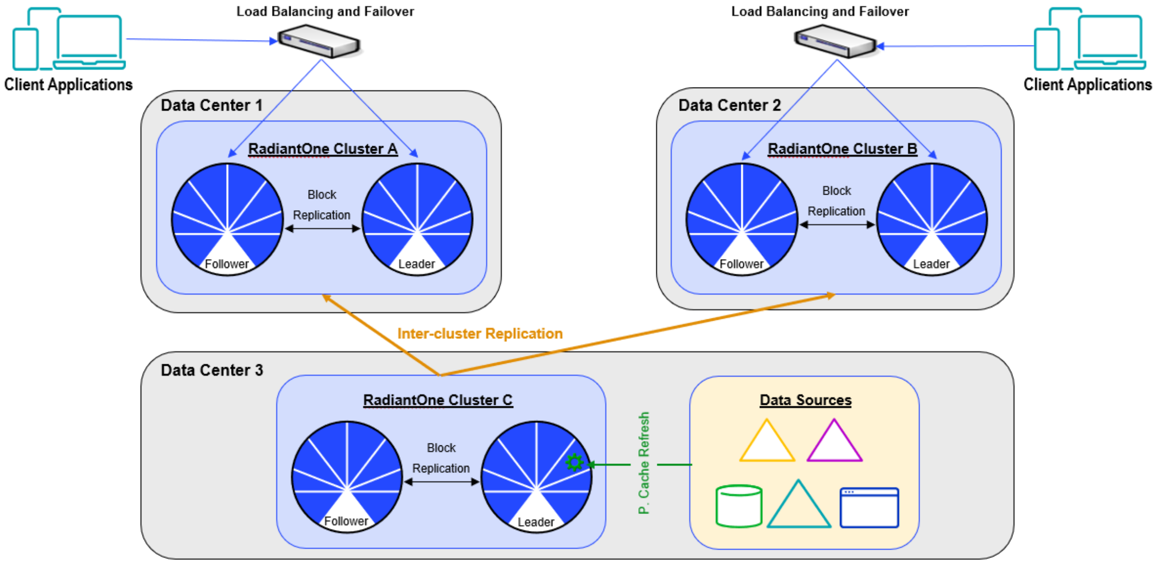
What level are password policies enforced (e.g. global, per group, per "ou"/tree branch, per user)?

What are the requirements of the policies themselves (e.g. password strength, lockout policy, password hash…etc.)?

Chapter 3: Import Data into RadiantOne Universal Directory

The recommended approach is to import the data as is (stick to the original DIT of the backend) to avoid complex re-mappings of group memberships. The import of the data is achieved through a persistent cache initialization of the proxy view. Once the data is in persistent cache, complex reorganizations of the original DIT can be done using virtualization. This includes things like flattening the hierarchy to get a list of users and groups, and merging overlapping users and groups (requiring correlation)…etc. Once you've configured the desired virtual view(s) as persistent cache, this image can be replicated to a RadiantOne Universal Directory (HDAP) store. This allows a separation of duties between the persistent cache refresh maintenance/process and the layer consumed by client applications. This also simplifies the cutover process once the backend server is fully decommissioned. The persistent cache refresh layer can be removed or repurposed.

Note – The persistent cache refresh (Cluster C) and client consumption layers (Custers A & B) shown below are depicting clusters containing two nodes for each. A Radiant Logic Architect can assess your throughput needs and recommend the best architecture.

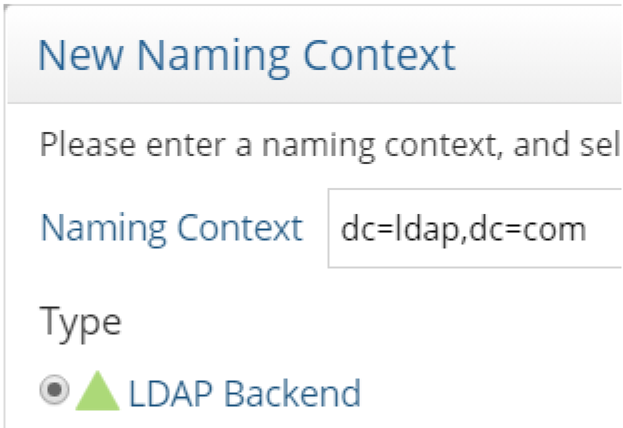


Note – if you would like to discuss a particular use case or alternate approach, please contact your Radiant Logic Account Representative.

To get the existing data, create a proxy view of the backend directory and create a persistent cache as outlined below. The terms Persistent Cache/Refresh Layer and Client Consumption layer are used below to describe the configuration applicable to each.

On the Persistent Cache/Refresh Layer:

1. Define an LDAP data source for the backend directory on the Main Control Panel -> Settings -> Server Backend -> LDAP Data Sources section.
2. Create a Root Naming Context from the Main Control Panel -> Directory Namespace tab of type “LDAP Backend”. If you need assistance, see the RadiantOne Namespace Configuration Guide.



New Naming Context

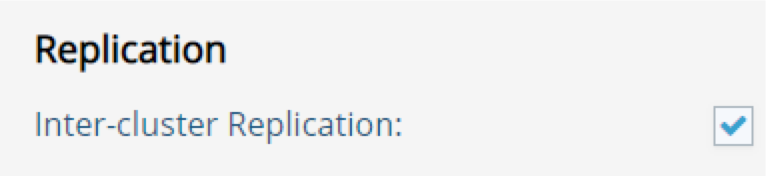
Please enter a naming context, and sel

Naming Context: dc=ldap,dc=com

Type

● LDAP Backend

3. Configure and initialize a persistent cache for the proxy view in addition to the desired refresh strategy (e.g. periodic or real-time). If you need assistance, see the RadiantOne Deployment and Tuning Guide.
4. (Optional) If you need to configure more advanced views/hierarchies, you can virtualize the persistent cache as an LDAP directory backend and create the desired view. Then, define a persistent cache for this view.
5. Enable Inter-cluster replication for the final persistent cache view that contains the image that should be replicated to the Client Consumption Layer RadiantOne Universal Directory (HDAP) store.



Replication

Inter-cluster Replication: ☑

6. From the Main Control Panel -> Settings -> Server Backend -> LDAP Data Sources section, verify the replicationjournal LDAP data source points to the desired journal.
7. Temporarily stop the persistent cache refresh (if it is running) and export the persistent cache view into an LDIF file. Copy this file to the Client Consumption layer machine.

On the Client Consumption Layer:

1. Configure the target RadiantOne Universal Directory (HDAP) store with the same root naming context as the backend directory (the one expected by client applications).
2. Initialize the RadiantOne Universal Directory (HDAP) store with the export of the persistent cache image.
3. Enable inter-cluster replication on the RadiantOne Universal Directory (HDAP) store. For assistance with this configuration, see the RadiantOne Namespace Configuration Guide.

Inter-cluster Replication: ☑

4. From the Main Control Panel -> Settings -> Server Backend -> LDAP Data Sources section, verify the replicationjournal LDAP data source points to the desired journal (and should be the same location referenced in the replicationjournal data source on the Persistent Cache/Refresh Layer – verified in step 6 above in the previous section).

As changes are detected on the backend legacy LDAP, the persistent cache views are refreshed and then replicated to the RadiantOne Universal Directory (HDAP) store that is being consumed by the client applications that have been migrated to the new directory. A few things to keep in mind:

- For bind operations, the persistent cache must contain the user passwords from the backend directory. The hashed passwords are then replicated to the RadiantOne Universal Directory (HDAP) store. As long as the password hash is compatible with the RadiantOne Universal Directory, users should be able to bind against it. Otherwise, binds need redirected. Consult with a Radiant Logic Architect so they can recommend the appropriate configuration.
- If client applications perform modifications, additional configuration is required to properly route the changes to the persistent cache/refresh layer. Consult with a Radiant Logic Architect so they can recommend the appropriate configuration.

Chapter 4: Configure RadiantOne Server Settings

Configure the appropriate server settings on the Client Consumption Layer machine:

LDAP Controls and Extensions

RadiantOne supports the following controls and extensions:

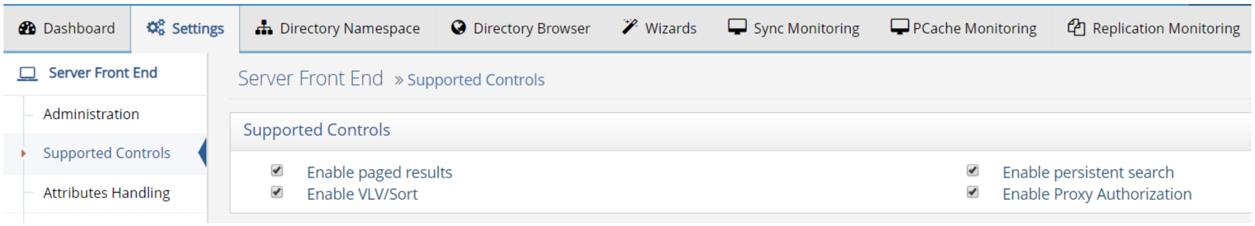
- Subtree Delete Control - 1.2.840.113556.1.4.805
- Password expired notification control - 2.16.840.1.113730.3.4.4
- Password expiring notification control - 2.16.840.1.113730.3.4.5
- Password policy control - 1.3.6.1.4.1.42.2.27.8.5.1
- Persistent search control - 2.16.840.1.113730.3.4.3
- Virtual list view request control - 2.16.840.1.113730.3.4.9
- Proxied authorization (version 2) control, described in RFC 4370 - 2.16.840.1.113730.3.4.18
- Server-side sort request, described in RFC 2891 - 1.2.840.113556.1.4.473
- Authorization bind identity response control, described in RFC 3829 - 2.16.840.1.113730.3.4.15
- Authorization bind identity request control, described in RFC 3829 - 2.16.840.1.113730.3.4.16
- Who Am I extended operation, described in RFC 4532 - 1.3.6.1.4.1.4203.1.11.3
- Paged Results Control - 1.2.840.113556.1.4.319
- Dynamic entries extension, described in RFC 2589 - 1.3.6.1.4.1.1466.101.119.1
- All Operational Attributes feature, described in RFC 3673 - 1.3.6.1.4.1.4203.1.5.1
- Absolute True and False Filters as described in RFC 4526 - 1.3.6.1.4.1.4203.1.5.3

The following controls that could be used in Sun Java Directory/ODSEE are not supported in RadiantOne:

- Manage DSA IT control, described in RFC 3296 - 2.16.840.1.113730.3.4.2
- Get effective rights request control - 1.3.6.1.4.1.42.2.27.9.5.2
- Account usability control - 1.3.6.1.4.1.42.2.27.9.5.8
- Specific backend search request control - 2.16.840.1.113730.3.4.14
- Real attributes only request control - 2.16.840.1.113730.3.4.17

- Virtual attributes only request control - 2.16.840.1.113730.3.4.19

Paged Results, VLV/Sort, Persistent Search and Proxy Authorization Controls are enabled from the Main Control Panel -> Settings tab-> Server front end -> Supported Controls. For details about each, please see the RadiantOne System Administration Guide.

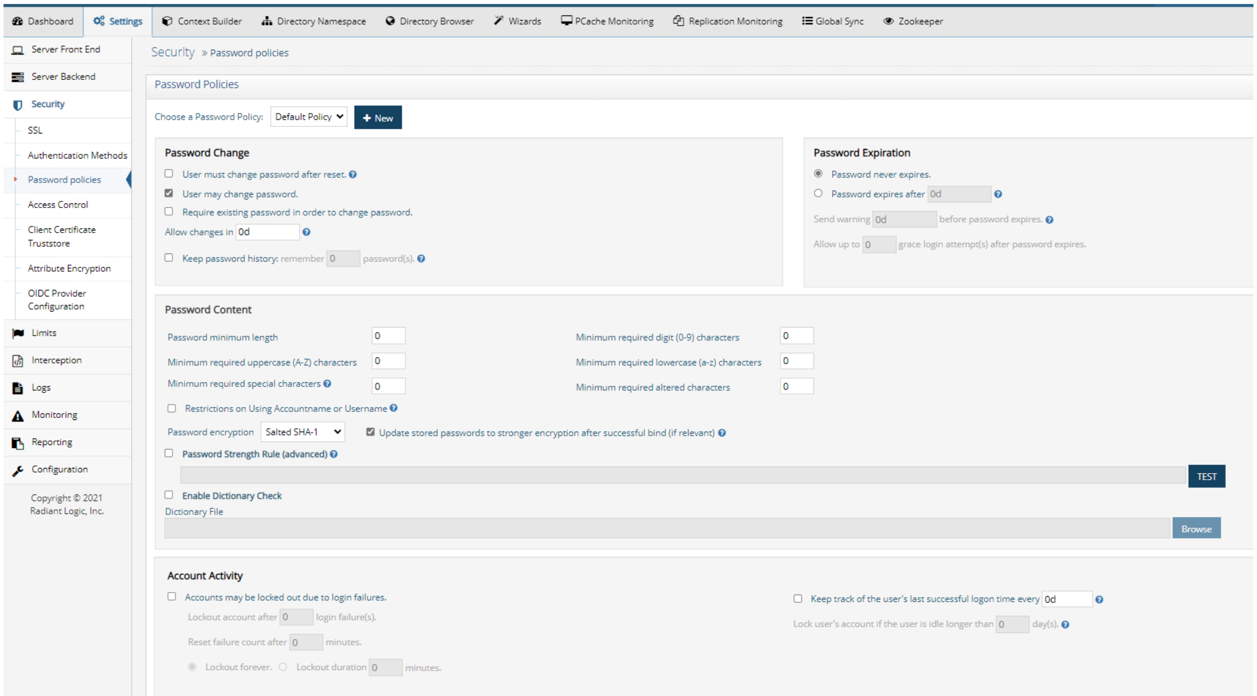


Server Front End » Supported Controls

Supported Controls

☑ Enable paged results
☑ Enable VLV/Sort
☑ Enable persistent search
☑ Enable Proxy Authorization

Password expired notification, password expiring notification, and password policy control are configured for password policies. Configure password policies on the Client Consumption layer.



Security » Password policies

Password Policies

Choose a Password Policy: Default Policy

Password Change

☐ User must change password after reset.
☑ User may change password.
☐ Require existing password in order to change password.
Allow changes in 0d
☐ Keep password history: remember 0 password(s).

Password Expiration

● Password never expires.
○ Password expires after 0d
Send warning 0d before password expires.
Allow up to 0 grace login attempt(s) after password expires.

Password Content

Password minimum length 0
Minimum required uppercase (A-Z) characters 0
Minimum required special characters 0
Minimum required digit (0-9) characters 0
Minimum required lowercase (a-z) characters 0
Minimum required altered characters 0
☐ Restrictions on Using Accountname or Username
Password encryption Salted SHA-1
☑ Update stored passwords to stronger encryption after successful bind (if relevant)
☐ Password Strength Rule (advanced)
☐ Enable Dictionary Check
Dictionary File

Account Activity

☐ Accounts may be locked out due to login failures.
Lockout account after 0 login failure(s).
Reset failure count after 0 minutes.
● Lockout forever. ○ Lockout duration 0 minutes.
☐ Keep track of the user's last successful logon time every 0d
Lock user's account if the user is idle longer than 0 day(s).

RootDSE

Directory Servers provide information about themselves to clients through the rootDSE. It contains information about the server in the form of attributes, some of which are multi-valued.

The rootDSE may contain information about the vendor, the naming contexts the server supports, the LDAP controls the server supports, the supported SASL mechanisms, schema location, and other information. The contents of the rootDSE generally determine the sequence and format of requests clients issue to the server.

The RadiantOne rootDSE is located at <RLI_HOME>\vds_server\conf\rootdse.ldif and is the default content returned to clients when they request the rootDSE (an LDAP search request with an empty DN). Some LDAP clients search the rootDSE to determine the naming contexts available in the LDAP directory and leverage this information to determine the baseDN (starting point in the directory) to pass in search requests.

Plugins

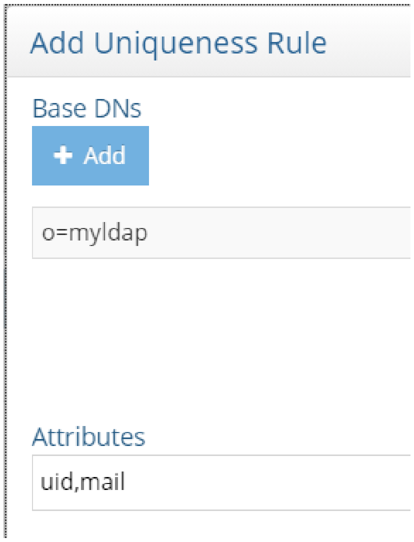
Some legacy LDAP directories support plug-ins to add specific functionality to the server.

Some of the most commonly used plugins and how to configure them in RadiantOne are described in this section.

Attribute Uniqueness

The Attribute Uniqueness plugin in legacy LDAP directories ensures that the value of a given attribute is unique among all entries of a subtree.

To enable comparable functionality in RadiantOne, from the Main Control Panel, navigate to the Setting tab -> Interception section (requires Expert Mode) -> Special Attributes Handling. Locate the Attribute Uniqueness setting and configure the attributes here.



Add Uniqueness Rule

Base DNs: o=myldap

Attributes: uid,mail

Referential Integrity

The referential integrity plug-in in legacy LDAP directories performs integrity updates on specified attributes immediately after a delete, rename, or move operation. It ensures that all attributes that reference the deleted, renamed or moved entry are updated accordingly.

To enable comparable functionality in RadiantOne, from Main Control Panel, navigate to the Setting tab -> Interception section (requires Expert Mode) -> Special Attributes Handling. Locate the Referential Integrity setting and configure the references here.

Linked Attributes

The isMemberOf plug-in in legacy LDAP directories enables clients to check a user's group membership by requesting the isMemberOf attribute in the user entries. This can be more efficient than searching in group entries looking for a uniquemember (especially in situations where group entries can be large/have many members).

To enable comparable functionality in RadiantOne, from Main Control Panel, navigate to the Setting tab -> Interception section (requires Expert Mode) -> Special Attributes Handling. Locate the Linked Attributes setting and configure the link between the location of users and the location of potential groups they are a member of here. RadiantOne computes isMemberOf only when the attribute is explicitly requested from clients.

This setting can be used for other back-link/forward-link attributes also (e.g. manager, owner, reportsTo…etc.).

Strong Password Check

The Strong Password Check plug-in enables the Directory Server to verify that a user's password doesn't contain unallowed strings from a specified dictionary file. This can be used as a method to enforce strong password policies.

To enable comparable functionality in RadiantOne, from Main Control Panel, navigate to the Setting tab ->Security -> Password policies. Locate the Password Content section and check the option to Enable Dictionary Check. Click Browse to navigate to the dictionary file.

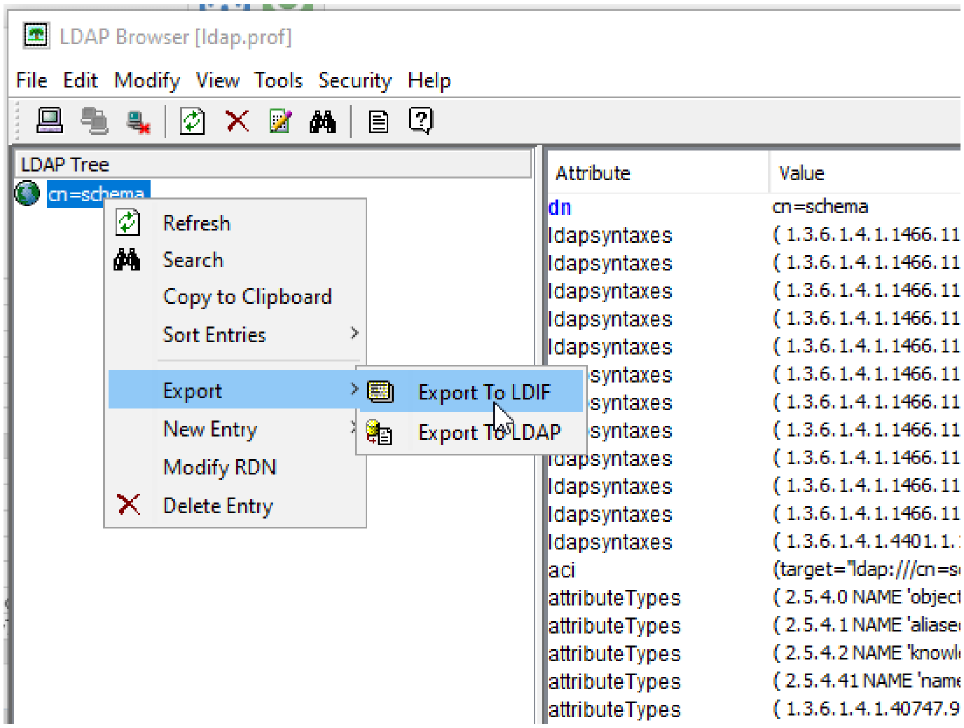
The dictionary file must be a text-formatted file containing one dictionary word per line.

Schema

The RadiantOne LDAP schema is comprised a series of LDIF files located: <RLI_HOME>\vds_server\conf\ldaschema_XX.ldif. XX being the number indicating the order in which the files are loaded. To extend the schema, the easiest approach is to get the object classes and attributes in LDIF format and then name the file ldapschema_XX.ldif where XX is the sequence you want the file loaded.

IMPORTANT NOTE - If you apply a new ldapschema_XX.ldif file and it has a number GREATER than 50 (e.g. ldapschema_51.ldif) and this definition includes object classes or attributes that are already defined in the VDS schema (in lower numbered schema files), the existing definitions are overridden with the latest definitions. This only starts AFTER the ldapschema_50.ldif file. Otherwise, the definition in the lower numbered files are not overridden.

If the LDAP directory stores the schema information in the cn=schema naming context, connect to this naming context from the RadiantOne LDAP Browser and you can export the schema to LDIF from there. Name the file ldapschema_XX.ldif (where XX is the sequence you want the schema loaded in) and save in <RLI_HOME>/vds_server/conf.



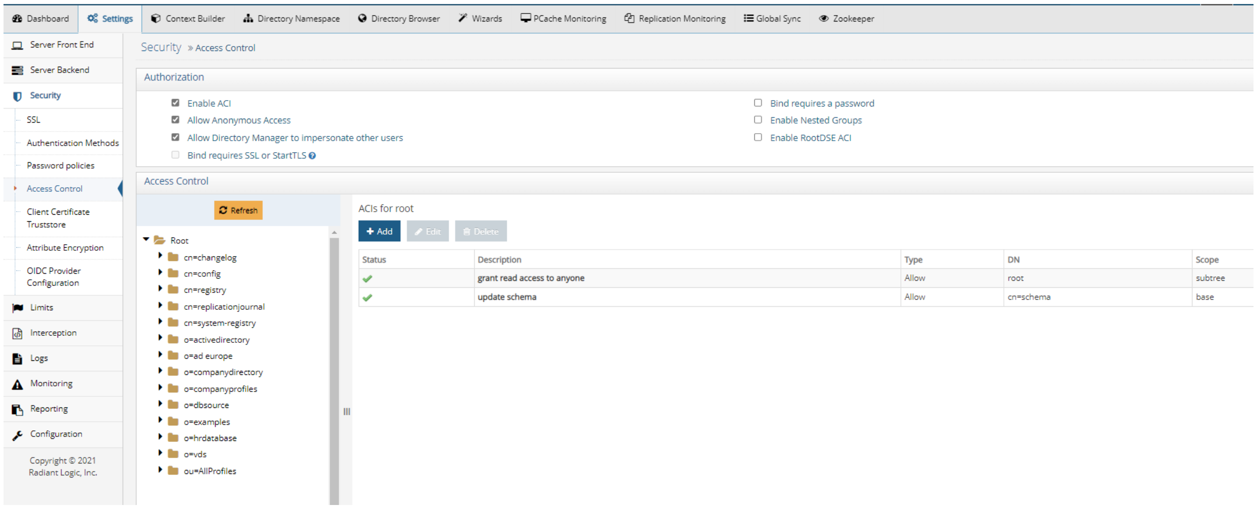
Attribute	Value
dn	cn=schema
ldapsyntaxes	(1.3.6.1.4.1.1466.11
ldapsyntaxes	(1.3.6.1.4.1.1466.11
ldapsyntaxes	(1.3.6.1.4.1.1466.11
ldapsyntaxes	(1.3.6.1.4.1.1466.11
ldapsyntaxes	(1.3.6.1.4.1.1466.11
syntaxes	(1.3.6.1.4.1.1466.11
syntaxes	(1.3.6.1.4.1.1466.11
syntaxes	(1.3.6.1.4.1.1466.11
ldapsyntaxes	(1.3.6.1.4.1.1466.11
ldapsyntaxes	(1.3.6.1.4.1.1466.11
ldapsyntaxes	(1.3.6.1.4.1.1466.11
ldapsyntaxes	(1.3.6.1.4.1.4401.1.
aci	(target="ldap:///cn=s
attributeTypes	(2.5.4.0 NAME 'object
attributeTypes	(2.5.4.1 NAME 'aliase
attributeTypes	(2.5.4.2 NAME 'knowl
attributeTypes	(2.5.4.41 NAME 'name
attributeTypes	(1.3.6.1.4.1.40747.9

ACLs

RadiantOne provides migration utilities to assist with translating the existing access controls into RadiantOne format. aciUtils and ibmAciMigration utilities are located in <RLI_HOME>/bin/advanced.

For details on using these utilities to migrate ACLs from the backend LDAP directory to RadiantOne Universal Directory (HDAP), see the RadiantOne ACI Migration Guide.

Access controls can be viewed and defined manually from the Main Control Panel -> Settings tab -> Security -> Access Controls section.



Authorization

☑ Enable ACI
☑ Allow Anonymous Access
☑ Allow Directory Manager to impersonate other users
☐ Bind requires SSL or StartTLS
☐ Bind requires a password
☐ Enable Nested Groups
☐ Enable RootDSE ACI

Access Control

- Root
  - cn=changelog
  - cn=config
  - cn=registry
  - cn=replicationjournal
  - cn=system-registry
  - o=activedirectory
  - o=ad europe
  - o=companydirectory
  - o=companyprofiles
  - o=dbsource
  - o=examples
  - o=hrdatabase
  - o=vds
  - ou=AllProfiles

ACIs for root

Status	Description	Type	DN	Scope
✓	grant read access to anyone	Allow	root	subtree
✓	update schema	Allow	cn=schema	base

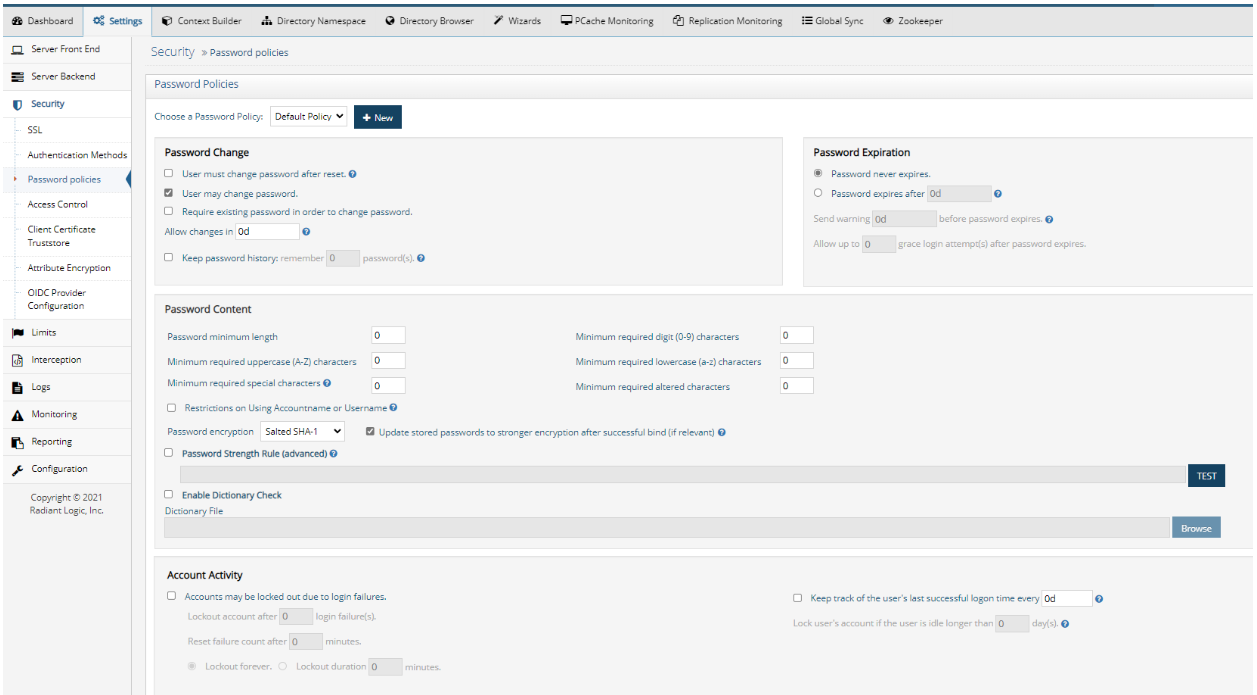
Password Policies

To support best practices around auditing and maintenance, RadiantOne only supports password policies assigned to LDAP groups or sub-trees (user's located in a given container in the FID namespace). Password policies defined at the user level are not supported. If you are replacing an LDAP directory that enforces password policies at the user level (e.g. in the passwordpolicysubentry attribute), when preparing the LDIF from the underlying directory (that you will use to initialize RadiantOne Universal Directory) do not include the passwordPolicySubentry attribute and move to use password policies defined at the group and/or "OU" (subtree) level.

Details about the RadiantOne password policy implementation are here:

https://tools.ietf.org/html/draft-behera-ldap-password-policy-10

The screen shot below shows the possible properties for RadiantOne Universal Directory password policies. For details on the properties see the RadiantOne System Administration Guide.



Password Policies

Choose a Password Policy: Default Policy

Password Change

- ☐ User must change password after reset.
- ☑ User may change password.
- ☐ Require existing password in order to change password.
- Allow changes in 0d
- ☐ Keep password history: remember 0 password(s).

Password Expiration

- ● Password never expires.
- ○ Password expires after 0d
- Send warning 0d before password expires.
- Allow up to 0 grace login attempt(s) after password expires.

Password Content

Password minimum length	0	Minimum required digit (0-9) characters	0
Minimum required uppercase (A-Z) characters	0	Minimum required lowercase (a-z) characters	0
Minimum required special characters	0	Minimum required altered characters	0

- ☐ Restrictions on Using Accountname or Username
- Password encryption: Salted SHA-1
- ☑ Update stored passwords to stronger encryption after successful bind (if relevant)
- ☐ Password Strength Rule (advanced)
- ☐ Enable Dictionary Check
- Dictionary File

Account Activity

- ☐ Accounts may be locked out due to login failures.
  - Lockout account after 0 login failure(s).
  - Reset failure count after 0 minutes.
  - ● Lockout forever. ○ Lockout duration 0 minutes.
- ☐ Keep track of the user's last successful logon time every 0d
  - Lock user's account if the user is idle longer than 0 day(s).

Chapter 5: Determine the Application Usage and Cutover Strategy

Determine Cutover Strategy

Generally all applications are not switched to use the new directory at the same time. There is a gradual migration of applications to point to the new directory. This allows application teams to migrate and test on their own schedule.

Likewise, the legacy LDAP directory isn't immediately switched off overnight. There is generally a temporary time period where both the legacy LDAP directory and the RadiantOne Universal Directory store must co-exist. This results in a required temporary synchronization process between the two.

The persistent cache refresh process is in charge of keeping the LDAP directory in sync with the target RadiantOne Universal Directory store. This configuration is outlined in Chapter 3.

Analyze Client Requests

Once applications are modified to point to the RadiantOne Universal Directory, analyze the <RLI_HOME>/vds_server/logs/vds_server.log to track the sequence of requests in an effort to determine what settings to tweak in RadiantOne.

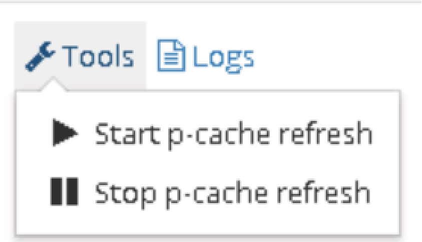
Items to pay special attention to:

- Does the client query for the rootDSE (blank base DN in the search request)? If so, what attributes are being requested?
- What are the requested attributes for searches? Are they sensitive attributes that require being stored encrypted?
- Do the requests invoke a sort control (special index for the attribute)?
- Do clients use proxy authorization (connect as a specific user and issue requests on behalf of someone else)?
- Do clients issue modification operations or only read operations? If modifications, which attributes are being modified?
- How is group membership checked (e.g. searching group entry if user is a member or searching user entry to see if they are a member of a certain group)?

Chapter 6: Decommission Legacy Directory

Once all applications have successfully migrated over to use RadiantOne Universal Directory, the legacy directory can be decommissioned and the persistent cache refresh and inter cluster replication processes can be stopped.

1. On the Client Consumption Layer, go to the Main Control Panel -> Directory Namespace tab.
2. Select the naming context for the RadiantOne Universal Directory (HDAP) store and on the right side, uncheck inter-cluster replication and click Save.
3. On the Persistent Cache Refresh Layer, go to the Main Control Panel -> PCache Monitoring tab.
4. Select the persistent cache refresh topology and once it loads, choose Tools -> Stop PCache Refresh.



For information on stopping RadiantOne components, see the RadiantOne Deployment and Tuning Guide.

For simpler scenarios, where a single layer is used (and clients are consuming the persistent cache view directly), the persistent cache can be converted to a RadiantOne Universal Directory store. This is a sensitive operation and must be performed during off-peak hours.

Before converting a persistent cache to an RadiantOne Universal Directory (HDAP) store, the persistent cache refresh should be stopped. You can set the refresh method to “none” on the Main Control Panel -> Directory Namespace -> Cache -> <cached branch> -> Refresh Settings tab. Also, suspend intercluster replication if it is used by setting “replicationInSuspendMode” : true, in ZooKeeper at /radiantone/<zk_version>/<clustername>/config/namings/<namingcontext_being_replicated>

Convert the persistent cache naming context into a RadiantOne Universal Directory with the following command using the <RLI_HOME>/bin/vdsconfig.bat(.sh) utility:

```
convert-pcache-to-hdap -namingcontext <namingcontext> [-instance <instance>]
```

Command Arguments:

-namingcontext <namingcontext>

[required] The name of the persistent cache naming context to be converted to a RadiantOne Universal Directory (HDAP) store.

-instance <instance>

The name of the RadiantOne FID instance. If this is not specified, the default instance named vds_server is used.

> **Note – before the conversion, you are prompted to confirm the operation. Enter “y” to confirm, or “n” to discontinue.**

After the persistent cache has been converted to a RadiantOne Universal Directory store, rebuild the index to remove any persistent cache operational attributes. For details on how to rebuild the index, see the RadiantOne Command Line Configuration Guide. If intercluster replication is used, enable it by setting “replicationInSuspendMode” : false, in ZooKeeper at /radiantone/<zk_version>/<clustername>/config/namings/<namingcontext_being_replicated>