HCL Portal

Guide to Integrating HCL Portal v8.5 with LDAP

Updated by:
Fernanda Gomes
HCL Portal Support Software Engineer

Previously created by:
Jason Wicker
HCL Portal Development Software Engineer

Table of Contents

Introduction

Applications commonly require HCL Portal to integrate with existing user repositories to enable authentication, authorization, and user management. LDAP (Lightweight Directory Access Protocol) is the most common user repository type.

In most organizations, different groups administer HCL Portal and the LDAP server. Integrating the two requires in-depth knowledge of both systems. This document guides HCL Portal administrators on obtaining the necessary information to integrate HCL Portal with existing LDAP servers.

The procedures in this guide are best suited for the initial, manual configuration of a development or test system. Tasks 1-9 (below) help you establish the appropriate integration points and property values. Tasks 10-13 (below) guide you through the integration itself. These procedures are based on tests with a stand-alone HCL Portal v8.5 on Linux, with IBM Tivoli Directory Server as an LDAP.

HCL Portal v8.5 introduced the Configuration Wizard to replace certain ConfigEngine tasks from previous versions. The panels in the Configuration Wizard may change over time. The knowledge obtained in Tasks 1-9 should carry over to future iterations of the Configuration Wizard. Tasks 10-13 may change slightly with future Configuration Wizard maintenance.

After validating the security configuration in lower-level environments (e.g. test), you may choose to automate configuring security in higher-level environments (e.g. production) rather than follow the manual processes in Tasks 10-13.

For general information about HCL Portal security, refer to Web security concepts and considerations for HCL Portal administrators, especially sections 2, 3, 4, and 6 which focus on aspects of LDAP integration.

Lab Environment

The procedures in this guide are based on the HCL Portal Product Documentation and tests in a lab with:

- **HCL Portal v8.5+CF06**
  - **Linux**
  - **DB2**
- **IBM Tivoli Directory Server / IBM Security Directory Server**
  - **Windows**
- **Apache Directory Studio**
  - **Windows**

Task 1 - Understand the initial security configuration and how to restore it

By default, HCL Portal uses the Virtual Member Manager (VMM) file repository as its initial user repository after installation. HCL recommends against using this file repository in a production environment.

If you mis-configure security such that HCL Portal functions improperly, you may restore the initial security configuration to recover.

Task 2 - Decide between a federated repository and stand-alone LDAP

Use a federated repository, if possible. The stand-alone LDAP user registry configuration is deprecated in HCL Portal 8.5. Also, a federated repository provides more functionality and flexibility than stand-alone LDAP. The remainder of this document focuses on configuring a federated repository. The general approach of verifying configuration properties with simple LDAP clients (Tasks 5-9 [below]) carries over to stand-alone LDAP configurations.

Task 3 - Get basic LDAP information

This task is available in worksheet format in the wiki.

Part A: Discussion Items

Ask the *LDAP administrator* for the following information. *Items 1-6 are required*. The remainder are optional, in that you may deduce them during the subsequent tasks or use the defaults in most environments. As a best practice, verify any items deduced by simple clients with your LDAP administrator prior to relying on them in your application, particularly in production environments.

- **Required:** LDAP type and version. For example, *IBM Security Directory Server v6.3.1*.
- **Required:** Fully qualified host name of the LDAP server. For example,

ldapserver.hcl.com.

- **Required:** LDAP port - *389* is the default non-SSL port and *636* is the default SSL port (TLS port) for LDAP, from the Internet Assigned Numbers Authority. If non-SSL, how is the network secured?
- **Required:** If HCL Portal must communicate with the LDAP over SSL, the LDAP administrator may provide an SSL certificate to enable this. Optionally, WebSphere Application Server can retrieve the certificate from the LDAP server itself.
- **Required:** Base distinguished name or names (DN) – This is the entry point for VMM into the LDAP. Any data that HCL Portal needs should reside under this node in the directory information tree (DIT) of the LDAP. Note that base DNs should not overlap in VMM realms. For example, *o=hcl,c=us*.
- **Required:** LDAP bind user and password – The LDAP administrator should provide the full DN of the bind user. The bind user is the identity that HCL Portal via underlying VMM uses to connect to the LDAP. For example, *cn=root*.
- Will HCL Portal access the LDAP server directly or through any load balancer, firewall, or proxy? If not directly, does the load balancer, firewall, or proxy impose any idle timeout or otherwise limit TCP connections made through it? Does the LDAP itself restrict the total number of TCP connections per application? Should the total number of TCP connections per application be restricted for LDAP performance reasons?
- If not through a load balancer, should VMM fail over to some backup LDAP server when the primary server is not available? If so, confirm that the backup is an exact replica, including universally unique identifier (UUID) values (RFC 3928).
- A detailed explanation of the access rights that the bind user has on nodes, users, and groups under the base DN. Note that VMM acts as the bind user during every interaction with the LDAP except for verifying a specific user's password during the login process. Specifically, you must establish whether the bind user can:
  - Create and delete users
  - Update existing users
  - Create and delete groups
  - Update existing groups
- What object class(es) defines users in this LDAP? Does the LDAP store all users under a specific node in its DIT?
  - Provide an LDIF of a sample user. See Task 7, #3 (below).
- What object class(es) defines groups in this LDAP? Does the LDAP store all groups under a specific node in its DIT?
  - Provide an LDIF of a sample group. See Task 7, #3 (below).
  - If the LDAP includes both static and dynamic groups, provide an LDIF of each.
  - Does the LDAP allow you to create empty groups? Or must any new group include at least one member?
- Does the LDAP implement a group membership attribute? If so, what is it and what is its scope? By *scope*, does it resolve direct groups only; direct and nested groups; or dynamic, direct, and nested groups? For example, the membership attribute in IBM Security Directory Server, *ibm-allGroups*, can resolve direct, nested, and dynamic groups.
- Does the LDAP implement dynamic groups? If so, what object class defines these? And what is the dynamic member attribute (which attribute stores the URL that defines the group)?
  - Note that dynamic groups generally impose a high performance overhead on both HCL Portal and the LDAP, so the LDAP will ideally resolve these to group

membership attribute values.

- What is the UUID and can it be searched externally? Reference Article KB0012799.
- Does the LDAP rely on LDAP referrals to serve any requests?

Federated repositories may encompass multiple LDAPs. If so, the items above may differ for each LDAP. Subsequent tasks focus on adding a single LDAP with a single base DN to the default federated repository.

Part B: Action Items

1. Check for the LDAP type and version from Part A, #1 (above) in the list of fully supported LDAP servers.
    - HCL provides best effort support for LDAPs that are not listed.

Task 4 - Establish the requirements of the application for the user repository

This task is available in worksheet format in the wiki.

Ask the *application architect* for the following information.

1. Does the level of support from Task 3, Part B, #1 (above) meet the requirements of the application?
2. Does the communications link between HCL Portal and LDAP (Task 3, Part A, #3 [above]) meet the security requirements of the application? Security hardened systems generally communicate with the LDAP over connections secured by SSL.
3. What LDAP attribute(s) will users log in with? Note that the value of this attribute must be unique for all users in the scope of a given VMM realm. This guide addresses the default realm only, which the base portal uses.
    - Refer to the LDIF (Task 7, #3 [below]) for attribute values. For example, *uid: user1*.
4. Will the application rely upon nested groups for access control? Dynamic groups?
5. Will the application create groups and/or manage group membership? Will the application create users in the LDAP? Confirm that the LDAP bind user has sufficient access rights for these requirements (Task 3, Part A, #9 [above]).
6. Does the application require access to all users and groups in the LDAP under the base DN or just certain ones? If just certain users/groups, how are they distinguished? For example, by some attribute value like *portalUser=true*? Or under some specific node in the DIT, like *dc=portalusers,dc=hcl,dc=com*?
7. Should the HCL Portal super-administrator user and group (e.g. *wpsadmin* and *wpsadmins*) reside in LDAP or in some other respository (e.g. default file repository)?
8. Will the application rely upon virtual portal(s)? If so, how should the user population be divided into realms? A *realm* is a set of users that can log in to a specific virtual portal. This guide addresses only the default realm, used by the base portal. However, if the user population from a single LDAP must be divided into realms, this will affect what base entries you should define per LDAP.
    - Note that the portal administrator (e.g. wpsadmin) must exist in every realm associated with any virtual portal. If your portal administrator is in an LDAP, plan your

base entries and realms accordingly.

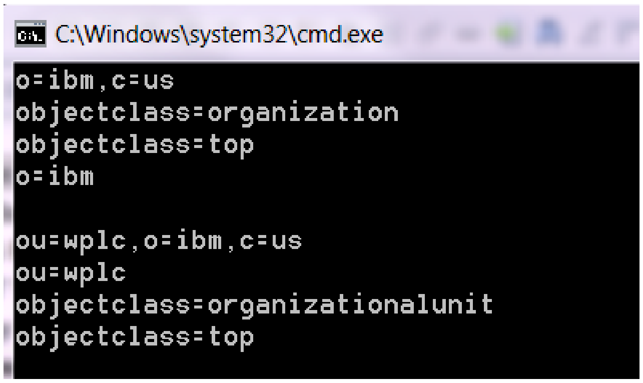
9. What user and group attribute values will the application need to read from or write to the LDAP?
10. Will the application need to manage any additional user attributes that are not available in the LDAP?

Task 5 - Test LDAP connectivity with ldapsearch

The Internet Engineering Task Force (IETF) specifies LDAP in RFC 4510 and related documents. Given this specification, you may reasonably assume that HCL Portal (via VMM) interacts with the LDAP similarly to simple clients, like ldapsearch and LDAP browsers. This and subsequent tasks guide you on using simple clients to test LDAP connectivity and to verify configuration properties.

Security tip: LDAP enables much security-related functionality, but is not an inherently secure protocol. ***Security hardened*** systems should communicate with the LDAP server only over connections secured by SSL. If non-SSL, you can capture packets with tools like tcpdump and **WireShark** to observe the LDAP requests and responses in this and subsequent tasks, as well as HCL Portal run-time. If non-SSL, double-check Task 4, #2 (above). You may choose to verify configuration properties with simple clients over non-SSL, then switch to SSL for HCL Portal configuration and run-time.

1. Determine whether you have access to an ldapsearch command line utility.
   - LDAP servers generally deliver ldapsearch.
   - Some operating systems provide ldapsearch.
   - If you do not already have ldapsearch, you may download this implementation.
   - If you prefer to use a graphical LDAP browser only, skip to Task 6.
2. Collocate ldapsearch with your HCL Portal server.
   - Your goal is to mimic, as closely as possible, how VMM will interact with the LDAP.
   - This may mean running ldapsearch from the HCL Portal server itself. Or it may mean running it from another system in the same subnet (e.g. in a DMZ).
   - Discuss your options with the network and system administrators.
   - If you choose to run ldapsearch from the HCL Portal server, consider whether to leave it on the system long-term. It could be useful for problem determination, but may not meet your security hardening requirements.
3. Verify the bind user (*cn=root*), bind password (*rootpw*), LDAP host name (*ldapserver.hcl.com*), port (*389*), and base DN (*o=hcl,c=us*)
   - ldapsearch -h *ldapserver.hcl.com* -p *389* –D *cn=root* -w *rootpw* -b "*o=hcl,c=us*" "objectclass=*"
   - These are example values. Refer to Task 3, Part A, #2-6 (above) for your actual values.
   - Output will look like:



```
o=ibm,c=us
objectclass=organization
objectclass=top
o=ibm

ou=wplc,o=ibm,c=us
ou=wplc
objectclass=organizationalunit
objectclass=top
```

- Most ldapsearch utilities support the syntax above. Refer to ldapsearch documentation if the command line returns any syntax errors.
- Some ldapsearch utilities require “-x” to force simple authentication.
- Refer to ldapsearch documentation if the LDAP requires clients to connect over SSL.
- If ldapsearch cannot connect, work with your LDAP and network administrators to verify the items from Task 3, Part A, #2-6 (above).

4. Make note of appropriate values for all of the following for your environment (**descriptors** correspond to Configuration Wizard fields in Task 10 [below]; use Worksheet_Tasks_5_to_9.pdf to keep track of your values):

- **LDAP repository ID**: *MyLDAP1*
  - This can be any ID you choose. It will uniquely identify the LDAP in the VMM configuration and WebSphere Application Server Integrated Solutions Console.
- **LDAP host name**: *ldapserver.hcl.com*
- **LDAP port**: *636*
- **Bind DN**: *cn=root*
- **Bind password**: *bindpassword*
- **Base DN**: *o=hcl,c=us*
- **Should WebSphere Portal use SSL for LDAP communications?**: *yes*

Task 6 - Test LDAP connectivity with an LDAP browser

LDAP browsers provide a graphical view of the directory information tree (DIT) of an LDAP server. Many also provide additional functions, such as exporting/importing LDIFs and constructing/executing searches.

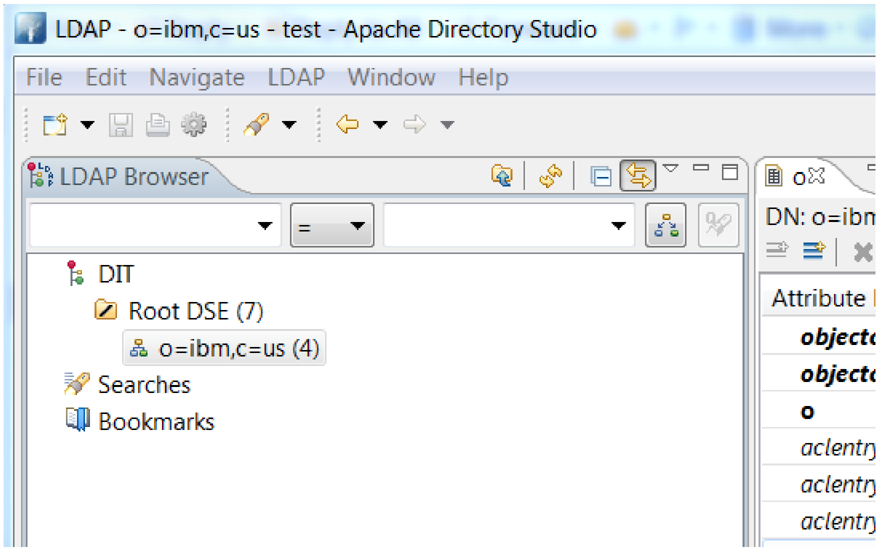
Connecting to an LDAP server with an LDAP browser verifies connectivity, in much the same way as ldapsearch did in Task 5. Subsequent tasks in this document use the LDAP browser to verify other aspects of the HCL Portal and LDAP integration.

Some LDAP servers deliver LDAP browsers. The examples in this and subsequent tasks use Apache Directory Studio. Refer to LDAP browser documentation for instructions on configuring the LDAP browser of your choice.

1.   Collocate the LDAP browser with the HCL Portal server.

- Your goal is to mimic, as closely as possible, how VMM will interact with the LDAP.
- This may mean running the LDAP browser from the HCL Portal server itself. Or it may mean running it from another system in the same subnet (e.g. DMZ).
- If you choose to run the LDAP browser from the HCL Portal server, consider whether to leave it on the system long-term. It could be useful for problem determination, but may not meet your security hardening requirements.
- Discuss your options with the network and system administrators.

2. The steps below use example values. Refer to Task 3, Part A, #2-6 (above) for your actual values.
3. Open **Apache Directory Studio** and select **LDAP > New Connection**.
4. In the **Network Parameter** panel, specify:
   - **Connection name** : *ldapserver connection name (your choice)*
   - **Hostname** : *ldapserver.hcl.com*
   - **Port** : *389*
   - **Encryption method** : *No encryption* (if you choose not to use SSL – additional configuration may be required if you choose to use SSL)
5. Click **Next**.
6. In the Authentication panel, specify:
   - **Authentication Method** : *Simple Authentication*
   - **Bind DN or user** : *cn=root*
   - **Bind password** : *rootpw*
7. Click **Check Authentication**. If authentication fails, test with ldapsearch. If the same fails with ldapsearch, work with your LDAP and network administrators to verify the items from Task 3, Part A, #2-6 (above).
8. Click **Next**.
9. In the **Browser Options** panel:
   - Un-check **Get Base DNs from Root DSE**
   - Specify **Base DN** : *o=hcl,c=us*
   - Specify **Referrals Handling** per your requirements (Task 3, Part A, #15 [above]). If you are not sure of your requirements, select **Ignore Referrals** at this time.
   - Check **Fetch operational attributes while browsing**. Among others, LDAPs generally consider UUIDs and membership attributes to be operational attributes.
10. Click **Finish**.
11. Confirm that you can browse the LDAP under your base DN.



12. Make note of appropriate values for all of the following for your environment (**descriptors** correspond to Configuration Wizard fields in Task 10 [below]; use Worksheet_Tasks_5_to_9.pdf to keep track of your values):

- **LDAP repository ID**: *MyLDAP1*
  - This can be any ID you choose. It will uniquely identify the LDAP in the VMM configuration and WebSphere Application Server Integrated Solutions Console.
- **LDAP host name**: *ldapserver.hcl.com*
- **LDAP port**: *636*
- **Bind DN**: *cn=root*
- **Bind password**: *bindpassword*
- **Base DN**: *o=hcl,c=us*
- **Should WebShere Portal use SSL for LDAP communications?**: *yes*

Task 7 - Verify how the LDAP represents entity types

For the purposes of HCL Portal, VMM distinguishes between two entity types. VMM represents users as entities of type *PersonAccount*. VMM represents groups as entities of type *Group*. In this task, you will verify how your LDAP represents users and groups.

1. Access the LDAP with an LDAP browser.
2. Navigate under the base DN (*o=hcl,c=us*) and locate a sample group and a sample user.
3. Export an LDIF of a sample user and of a sample group for future reference. To do so in Apache Directory Studio:
   - **Right-click on the record** in the DIT and select **Export > LDIF Export …**
   - For Returning Attributes, check both **All user attributes** and **Operational attributes**.
   - Click **Next**.
   - Specify the target **LDIF file**
     - *user1.ldif (see sample attached in wiki)*
     - *group1.ldif (see sample attached in wiki)*
   - Click **Finish**.
   - *Alternatively, get these LDIFs from the LDAP adminsitrator or from ldapsearch output.*

However, note that LDIFs and ldapsearch ***do not export operational attributes****, by default.*

4. Compare the object classes in these records (either directly in the LDAP browser or in the LDIFs). Which object class(es) defines groups in the LDAP? Which defines users? For example, consider:
   - usersample1 has objectClass=*top* and objectClass=*inetOrgPerson*
   - groupsample1 has objectClass=*top* and objectClass=*groupOfUniqueNames*
   - You cannot use the object class *top* to distinguish between these entity types.
   - Use the object class *inetOrgPerson* to distinguish users from other types of records in this LDAP.
   - Use the object class *groupOfUniqueNames* to distinguish groups from other types of records in this LDAP.
   - Note that some LDAPs have multiple types of groups (static, dynamic) and so may have multiple object classes to distinguish between these types.
5. Are all of the users under a single node in the DIT of the LDAP? What is the most specific node under which all users reside? Likewise, for groups. This may be determined trivially by browsing and visual inspection. If not, consider using a procedure like this example:
   - In Apache Directory Studio, **right-click the base DN** (*o=hcl,c=us*) and select **New > New Search ...**
   - Confirm that the **Search Base** is your base DN.
   - For **Filter**, enter **(objectClass=*groupOfUniqueNames*) –** Substitute the actual object class from Task 7, #4 (above).
   - For **Scope**, select **Subtree**.
   - Click **Search**.
     - *Alternatively, use ldapsearch and specify this base DN (-b) and search filter (final command-line argument).*
   - Review the search results (list of DNs) to determine the most specific node in the DIT that contains all groups to which your application requires access (from Task 4, #6 [above]).
   - For example, consider this list of DNs and requirements:
     - cn=group1,ou=groups,o=hcl,c=us – HCL Portal needs access to this group.
     - cn=group2,ou=orgA,ou=groups,o=hcl,c=us - HCL Portal needs access to this group.
     - cn=group3,ou=orgB,ou=groups,o=hcl,c=us - HCL Portal needs access to this group.
     - cn=ldapadmins,o=hcl,c=us – HCL Portal will not use this group.
     - The most specific DIT node under which you can find all HCL Portal groups is *ou=groups,o=hcl,c=us*
   - Repeat the search under *ou=groups,o=hcl,c=us* and confirm that the LDAP returns all requisite groups.
6. Create a sample user and group, if your application requires it (Task 4, #5 [above]).
   - Navigate to the location in the LDAP where users reside. In this example, *ou=users,o=hcl,c=us*.
   - **Single-click one of the users** in the DIT.
   - Select **LDAP > New Entry.**
   - Select **Use existing entry as template** (*uid=user1,ou=users,o=hcl,c=us*).
   - Click **Next**.

   - Verify the object classes are correct. In this example, make sure this list includes *inetOrgPerson*.
   - Click **Next**.
   - Verify the RDN and specify a value. In this example, *uid* could be set to *user2*.
   - Click **Next**.
   - Click **Finish**.

   1. If this step fails, verify that the bind user has sufficient access rights to create users in this LDAP.
   2. Repeat this process for a sample group.

7. Make note of appropriate values for all of the following for your environment (**descriptors** correspond to Configuration Wizard fields in Task 10 [below]; use Worksheet_Tasks_5_to_9.pdf to keep track of your values):
   - **LDAP group object classes**: *groupOfUniqueNames*
   - **LDAP group object classes for creating groups**: *groupOfUniqueNames*
   - **LDAP group search bases**: *ou=groups,o=hcl,c=us*
   - **LDAP PersonAccount object classes**: *inetOrgPerson*
   - **LDAP PersonAccount object classes for creating users**: *inetOrgPerson*
   - **LDAP search bases for the PersonAccount**: *ou=users,o=hcl,c=us*
   - **Group RDN attribute**: *cn*
   - **PersonAccount RDN attribute**: *uid*

Task 8 - Verify how the LDAP denotes group membership

LDAP groups enable administrative efficiency by allowing you to assign roles to groups of users rather than to each individual user. LDAPs represent group membership in several, sometimes redundant, ways. Refer to Web Security Concepts and Considerations for more information on static (direct and nested) groups, dynamic groups, and group membership attributes.

HCL Portal determines group membership for a given user when that user logs in. You must correctly configure both HCL Portal and VMM to ensure valid group membership determination.

Part A: What members constitute a given static group?

1. Browse the LDAP and navigate to a sample static group or inspect *group1.ldif* from Task 7, #3 (above).
2. Which attribute identifies the members of this group?
   - The value(s) for this attribute will be DNs of other users and/or groups in the LDAP. You may trivially identify the attribute by visual inspection. If not, then:
   - Refer to the object class definition or LDAP administrator for guidance. LDAP server documentation generally provides this. Many common object class definitions are also available on the web.
   - For example, objects of class *groupOfUniqueNames* store members as values of the attribute, *uniqueMember*.

    - Your LDAP may provide more than one groupMemberName attribute – see Part B, #3 (below) (e.g. uniqueMember and ibm-allMembers).
    - From a sample LDIF, here is a group with two options for groupMemberName:
        - dn: cn=group1,cn=groups,O=HCL,C=US
        - objectClass: groupOfUniqueNames
        - uniquemember: uid=user1,cn=users,O=HCL,C=US
        - ibm-allMembers: uid=user1,cn=users,O=HCL,C=US

3. Make note of appropriate values for all of the following for your environment (**descriptors** correspond to Configuration Wizard fields in Task 10 [below]; use Worksheet_Tasks_5_to_9.pdf to keep track of your values):
        - **Group member attribute**: *uniqueMember* or *ibm-allMembers*
        - **Group object class**: *groupOfUniqueNames*

Part B: Can groups be nested in this LDAP? And does the groupMemberName attribute include nested members?

1. Browse the LDAP and inspect the static groups or inspect *group1.ldif* (and similar) from Task 7, #3 (above).
2. Are any groups members of other groups? Or does the LDAP allow them to be?
    - If the LDAP stores users and groups under different nodes (searchBases from Task 7, #7 [above]), you could differentiate between users and groups by just inspecting the DNs stored in the groupMemberName attribute from Part A.
    - Nesting groups is common in LDAPs. *If the application architect and LDAP administrator cannot provide clear guidance, assume that HCL Portal must determine nested group membership.*
3. Does the groupMemberName attribute include direct members only, or direct and nested members?
    - Generally, these include direct members only.
    - If you found any nested groups in (2), you can check this. Consider, for example:
        - groupA has one member, groupB
        - groupB has one member, user1
        - Therefore, the scope of the groupMemberName attribute is *direct*. If the scope were nested, groupA would also list user1 as a member. HCL Portal may need to do some work at run-time to determine that user1 is a (nested) member of groupA.
    - Refer to the object class definition and LDAP documentation for more information.
    - Some LDAPs provide operational attributes to resolve nested group membership. For example, IBM Tivoli Directory Server provides ibm-allMembers. LDAPs do not export operational attributes to LDIFs, by default, so you may need to browse or specifically query the operational attributes in LDAP to determine this.
    - If your LDAP provides such an operational attribute, leveraging it could improve HCL Portal performance by avoiding the work of determining nested group membership when users log in and/or when an administrator requests membership of a given group.
    - For this example, the LDIF would look like:
        - dn: cn=groupA,cn=groups,O=HCL,C=US
        - objectClass: groupOfUniqueNames
        - uniquemember: cn=groupB,cn=groups,O=HCL,C=US
        - ibm-allMembers: cn=groupB,cn=groups,O=HCL,C=US

        - ibm-allMembers: uid=user1,cn=users,O=HCL,C=US

4. Make note of appropriate values for all of the following for your environment (**descriptors** correspond to Configuration Wizard fields in Task 10 [below]; use Worksheet_Tasks_5_to_9.pdf to keep track of your values):
    - **Group member attribute scope** =
        - *direct* (if using uniqueMember as the group member attribute)
        - *nested* (if using ibm-allMembers as the group member attribute)

Part C: Is a membership attribute available in this LDAP?

1. Browse a static group or inspect static group1.ldif from Task 7, #3 (above).
2. Identify one member of the sample group that is a user. For this example, consider that the user in *user1.ldif* is a member of the group in *group1.ldif*.
3. Inspect *user1.ldif* (only if the LDIF includes operational attributes – recall that you exported operational attributes in Task 7, #3 [above]).
    - It is important to refer to the LDIF at this stage. While testing this guide, the author found that Apache Directory Studio v2.0 did not display the operational attribute ibm-allGroups unless specifically requested, regardless of the configuration setting for displaying operational attributes.
4. Is there an attribute that lists all of the groups that the user is a member of?
    - For example, in IBM Security Directory Server, *ibm-allGroups*.
5. Confirm with the LDAP administrator, in LDAP documentation, or by example, the scope of any membership attribute.
    - Do the membership attribute values include only the groups that the user is a direct member of?
    - Do the membership attribute values include nested groups? For example, *user1* is a member of *groupsub1*. And *groupsub1* is a member of *groupsuper1*. Does *user1*'s membership attribute values include *groupsuper1*? If so, the membership attribute scope is **at least** *nested.*
6. **Make** note of appropriate values for all of the following for your environment (**descriptors** correspond to Configuration Wizard fields in Task 10 [below]; use Worksheet_Tasks_5_to_9.pdf to keep track of your values):
        - **Membership attribute name** = *ibm-allGroups*
        - **GC member attribute scope** = *nested*

Part D: Does this LDAP include dynamic groups?

1. Refer to Task 3, Part A, #13 (above) for whether the LDAP contains dynamic groups. If so, refer to Task 3, Part A, #13 for the object class that defines dynamic groups. For example, *groupofurls*. Otherwise, skip this (Task 8, Part D).
    - Ensure that the value of **Group object class** includes the dynamic group object class as well as the object class(es) for static groups.
2. Browse the LDAP and locate a sample dynamic group, if any exists. If you know the object class but not the location, search the base DN. For example, use a search filter of (objectclass=*groupofurls*).
3. Verify the name of the dynamic member attribute. For example, *memberurl*.
    - The value of this attribute will be a URL that essentially defines an LDAP search. The LDAP considers that the results of this search are the members of the dynamic group.
4. Determine the members of the sample dynamic group:
    - Execute the search specified by the dynamic member attribute. For example:
        - Given a dynamic group, *USEmployees*, with:

        - *memberurl*=ldap:///o=hcl,c=us??sub?(c=us)
        - The search would be like:
        - ldapsearch -h *ldapserver.hcl.com* -p *389* –D *cn=root* -w *rootpw* -b "*o=hcl,c=us*" "*c=us*"

5. Repeat Task 8, Part C, #2-4 (above) for this sample member to determine if a membership attribute resolves dynamic group membership.
    - If the membership attribute resolves all of dynamic, direct, and nested group membership then (**descriptors** correspond to Configuration Wizard fields in Task 10 [below]; use Worksheet_Tasks_5_to_9.pdf to keep track of your values):
        - **Group member attribute scope** = all
    - If the membership attribute does not resolve dynamic group membership, leave **Group member attribute scope** as its current value.
6. If the LDAP contains dynamic groups, but if the LDAP does not resolve dynamic groups to group membership attribute values, then proceed with LDAP integration but note that some additional manual configuration will be required after the automated integration task completes (Task 12, #3 [below]). Refer to the HCL Product Documentation section specific to your operating system and cluster configuration. Again, be aware of the performance overhead of using dynamic groups and test your application accordingly.

Task 9 - Final preparations before initial integration

Run these several tests and verify values for these several properties.

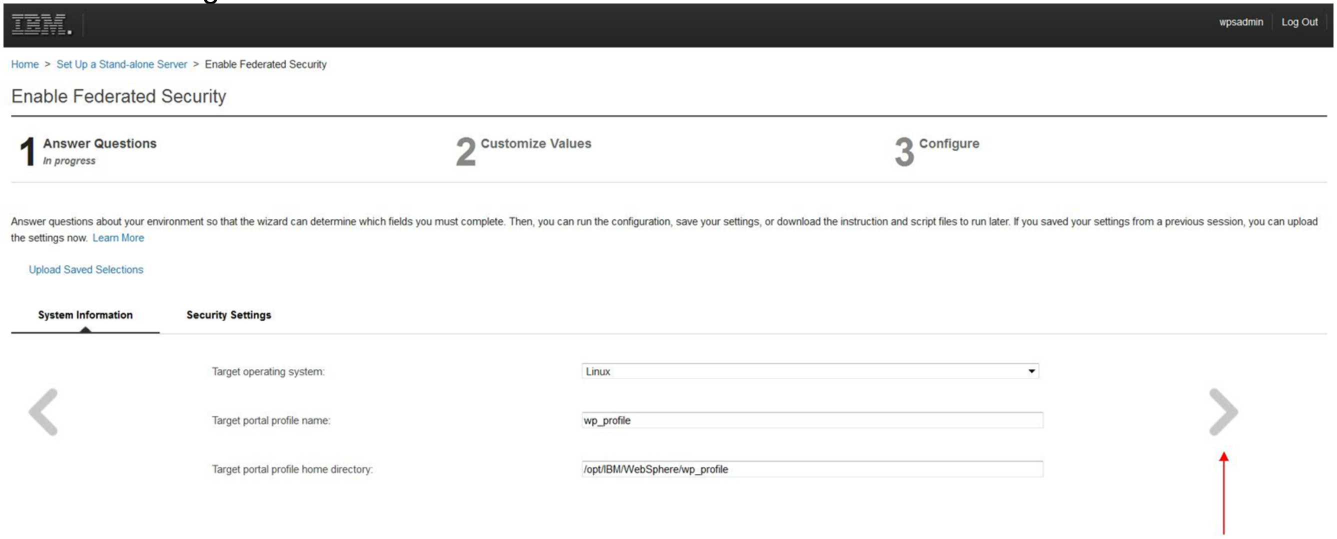
1. Refer to Task 4, #3 (above) and verify the properties for login. The Configuration Wizard will set the login property to *uid*. If your application requires another login property or additional login properties, take note and revisit this in Task 11 (below). Make note of the login properties in Worksheet_Tasks_5_to_9.pdf as **Federated repository properties for login**.
2. Refer to Task 3, Part A, #11 (above). If the LDAP requires any newly created group to include at least one member, make note of the dummy member:
    - **Group dummy member**=*uid=dummy*
3. Imitate how VMM searches for users when they attempt to log in. If this fails, correct configuration properties in Worksheet_Tasks_5_to_9.pdf. For example, use ldapsearch like:
    - ldapsearch -h *ldapserver.hcl.com* -p *389* -D *cn=root* -w *rootpw* -b "*ou=users,o=hcl,c=us*" "*(&(uid=user1)(objectclass=inetOrgPerson)*"
    - Ensure that the LDAP returns only one user record to confirm that the login attribute value is unique to this sample user.
4. If your LDAP supports such operations, optionally verify the sample user's password:
    - ldapsearch -h *ldapserver.hcl.com* -p *389* -D "*uid=user1,ou=users,o=hcl,c=us*" -w *user1password* -b "*uid=user1,ou=users,o=hcl,c=us*" "(objectClass=*)"
    - The goal of this command is to do an ldapbind with a sample user. The syntax above was chosen to avoid any access control problems in LDAP since the user is accessing its own record.
    - Some ldapsearch utilities require "-x" to force simple authentication.
5. Verify that the LDAP provides a unique UUID for the sample user and the bind user can access its value.
    - Refer to Task 3, Part A, #14 (above) for the UUID attribute name.
    - Is this UUID attribute name the default that VMM uses, per LDAP type?

   - If your LDAP administrator does not know the UUID attribute name, use the LDAP browser to look for the UUID attributes that VMM uses, by default, per LDAP type:
     - ibm-entryuuid
     - objectguid
     - dominounid
   - ldapsearch -h *ldapserver.hcl.com* -p *389* -D *cn=root* -w *rootpw* -b "*uid=user1,ou=users,o=hcl,c=us*" "(objectClass=*)" *ibm-entryuuid*
   - For example:
     - dn: uid=user1,cn=users,O=HCL,C=US
     - ibm-entryUuid: ab6a2ad3-e6d7-48a5-a9bb-136e60a58bdb
6. Given the UUID value from Task 9, #6 (above) verify that the LDAP supports searching based on UUID values.
   - ldapsearch -h *ldapserver.hcl.com* -p *389* -D *cn=root* -w *rootpw* -b "*o=hcl,c=us*" "(*ibm-entryuuid=ab6a2ad3-e6d7-48a5-a9bb-136e60a58bdb)”*
7. Repeat Task 9, #6-7 (above) for a sample group.

Task 10 - Integrate HCL Portal with LDAP – The Basics

In the preceding tasks, you used simple clients to verify the property values to specify in the Configuration Wizard panels. In this task you will add the LDAP to the federated repository (recommended, from Task 2 [above]). To do so, follow this procedure:

1. Run **backupConfig** to back up the WebSphere Application Server configuration.
2. Verify that server1 is running and access the Config Wizard on its internal port, per the HCL Product Documentation.
3. Choose **Set up a …** for your server type.
4. Choose **Enable federated security**.
5. Answer questions in the first step of the workflow according to your requirements, then click the right-arrow:



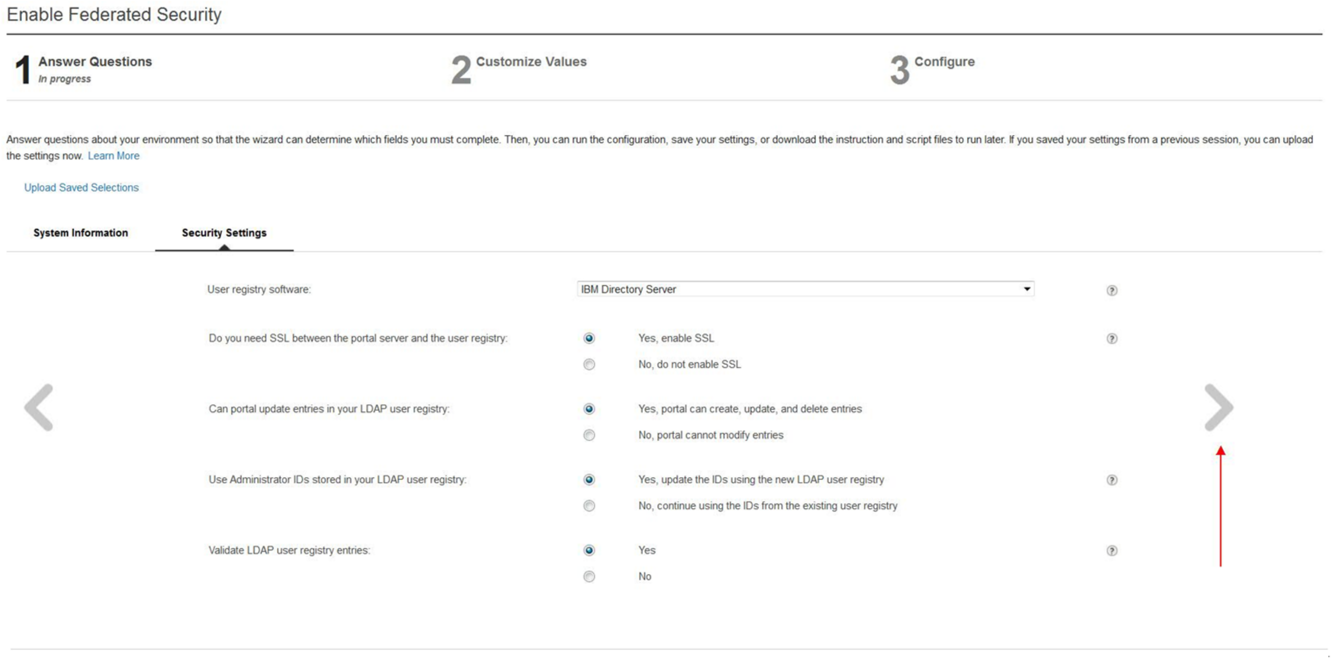
Enable Federated Security

Answer questions about your environment so that the wizard can determine which fields you must complete. Then, you can run the configuration, save your settings, or download the instruction and script files to run later. If you saved your settings from a previous session, you can upload the settings now.

Target operating system: Linux

Target portal profile name: wp_profile

Target portal profile home directory: /opt/IBM/WebSphere/wp_profile



Enable Federated Security

1 Answer Questions *In progress* | 2 Customize Values | 3 Configure

Answer questions about your environment so that the wizard can determine which fields you must complete. Then, you can run the configuration, save your settings, or download the instruction and script files to run later. If you saved your settings from a previous session, you can upload the settings now. Learn More

Upload Saved Selections

System Information | Security Settings

User registry software: IBM Directory Server

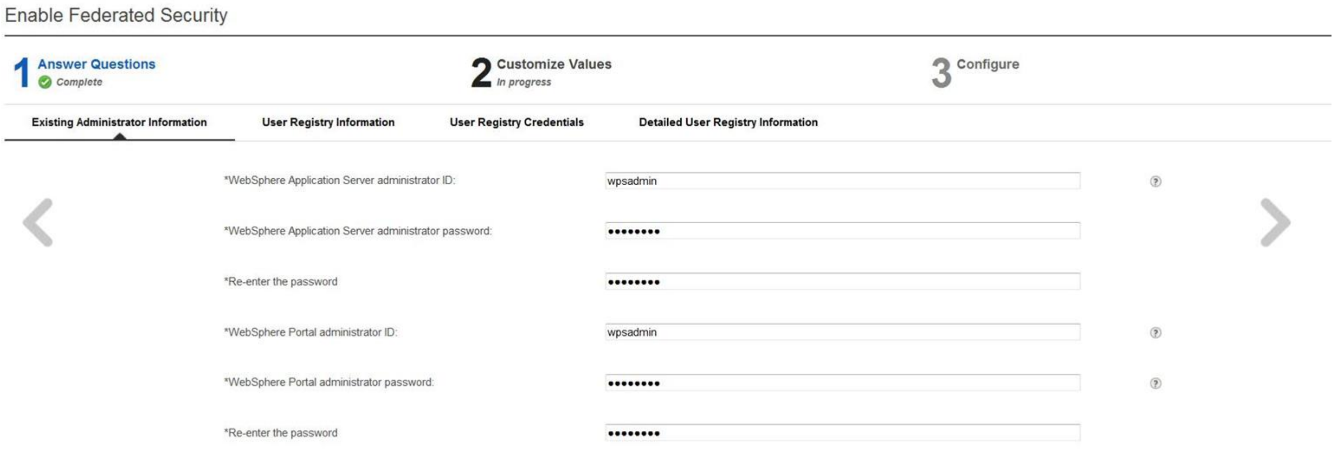
Do you need SSL between the portal server and the user registry:
● Yes, enable SSL
○ No, do not enable SSL

Can portal update entries in your LDAP user registry:
● Yes, portal can create, update, and delete entries
○ No, portal cannot modify entries

Use Administrator IDs stored in your LDAP user registry:
● Yes, update the IDs using the new LDAP user registry
○ No, continue using the IDs from the existing user registry

Validate LDAP user registry entries:
● Yes
○ No

6. In the second step of the work flow, on the Existing Administrator Information panel, provide the initial administrative IDs and passwords that were specified during installation, then click the right-arrow:



Enable Federated Security

1 Answer Questions *Complete* | 2 Customize Values *In progress* | 3 Configure

Existing Administrator Information | User Registry Information | User Registry Credentials | Detailed User Registry Information

*WebSphere Application Server administrator ID: wpsadmin

*WebSphere Application Server administrator password: ••••••••

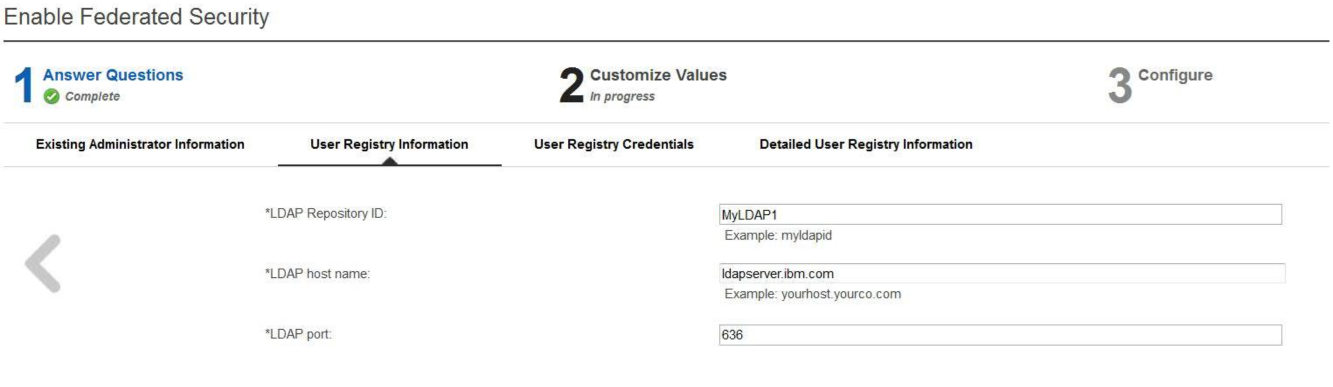
*Re-enter the password ••••••••

*WebSphere Portal administrator ID: wpsadmin

*WebSphere Portal administrator password: ••••••••

*Re-enter the password ••••••••

7. On the User Registry Information panel, provide the LDAP identifier, host name, and port, then click the right-arrow:

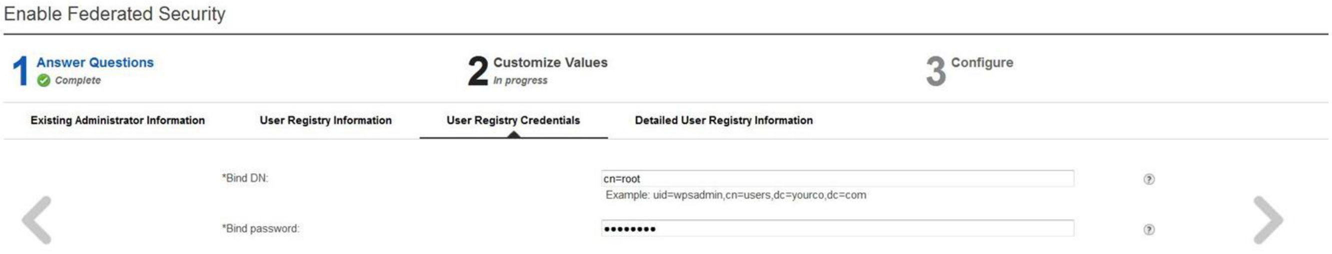

Enable Federated Security
1 Answer Questions
Complete
2 Customize Values
In progress
3 Configure
Existing Administrator Information
User Registry Information
User Registry Credentials
Detailed User Registry Information

*LDAP Repository ID: MyLDAP1
Example: myldapid

*LDAP host name: ldapserver.ibm.com
Example: yourhost.yourco.com

*LDAP port: 636

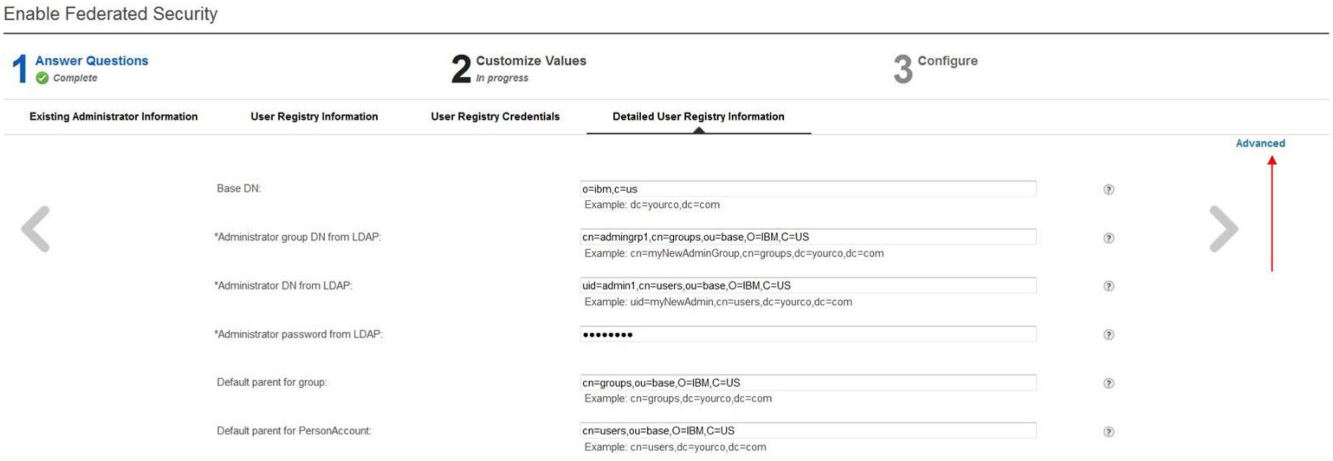
8. In the User Registry Credentials panel, provide the bind DN and password, then click the right-arrow:


Enable Federated Security
1 Answer Questions
Complete
2 Customize Values
In progress
3 Configure
Existing Administrator Information
User Registry Information
User Registry Credentials
Detailed User Registry Information

*Bind DN: cn=root
Example: uid=wpsadmin,cn=users,dc=yourco,dc=com

*Bind password: ••••••••

9. In the Detailed User Registry Information panel, provide the detailed information for adding this LDAP:


Enable Federated Security
1 Answer Questions
Complete
2 Customize Values
In progress
3 Configure
Existing Administrator Information
User Registry Information
User Registry Credentials
Detailed User Registry Information
Advanced

Base DN: o=ibm,c=us
Example: dc=yourco,dc=com

*Administrator group DN from LDAP: cn=admingrp1,cn=groups,ou=base,O=IBM,C=US
Example: cn=myNewAdminGroup,cn=groups,dc=yourco,dc=com

*Administrator DN from LDAP: uid=admin1,cn=users,ou=base,O=IBM,C=US
Example: uid=myNewAdmin,cn=users,dc=yourco,dc=com

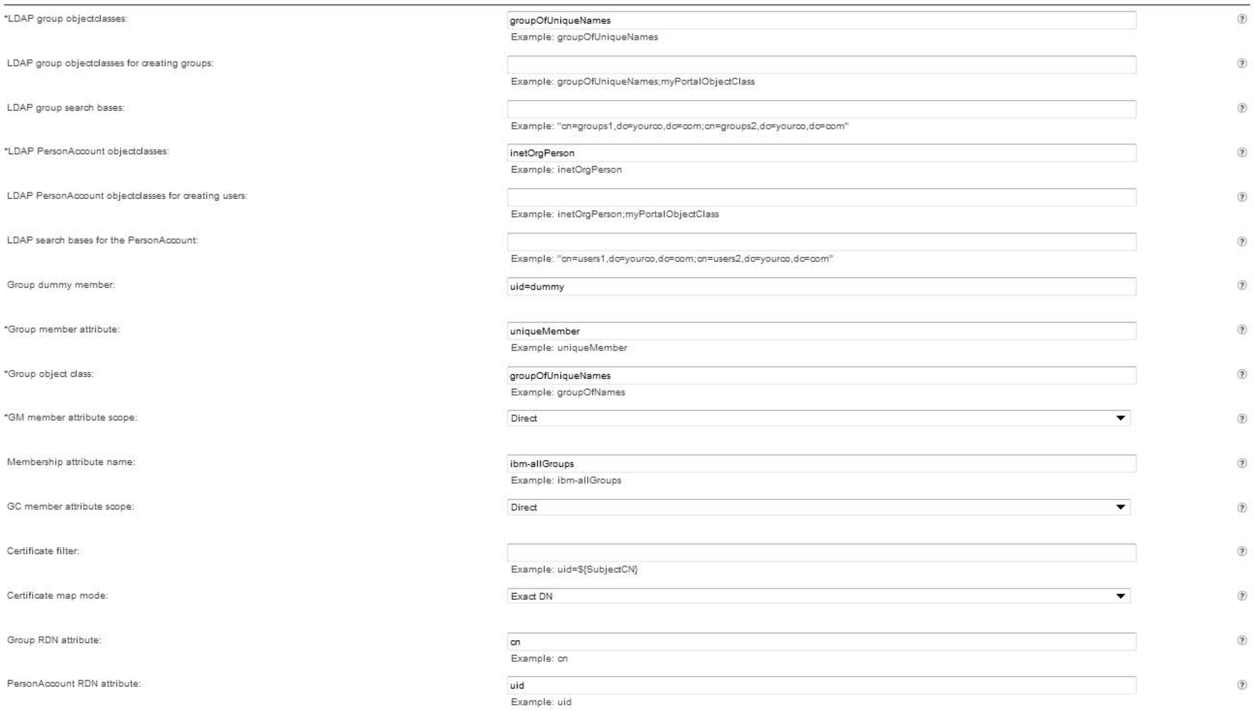
*Administrator password from LDAP: ••••••••

Default parent for group: cn=groups,ou=base,O=IBM,C=US
Example: cn=groups,dc=yourco,dc=com

Default parent for PersonAccount: cn=users,ou=base,O=IBM,C=US
Example: cn=users,dc=yourco,dc=com

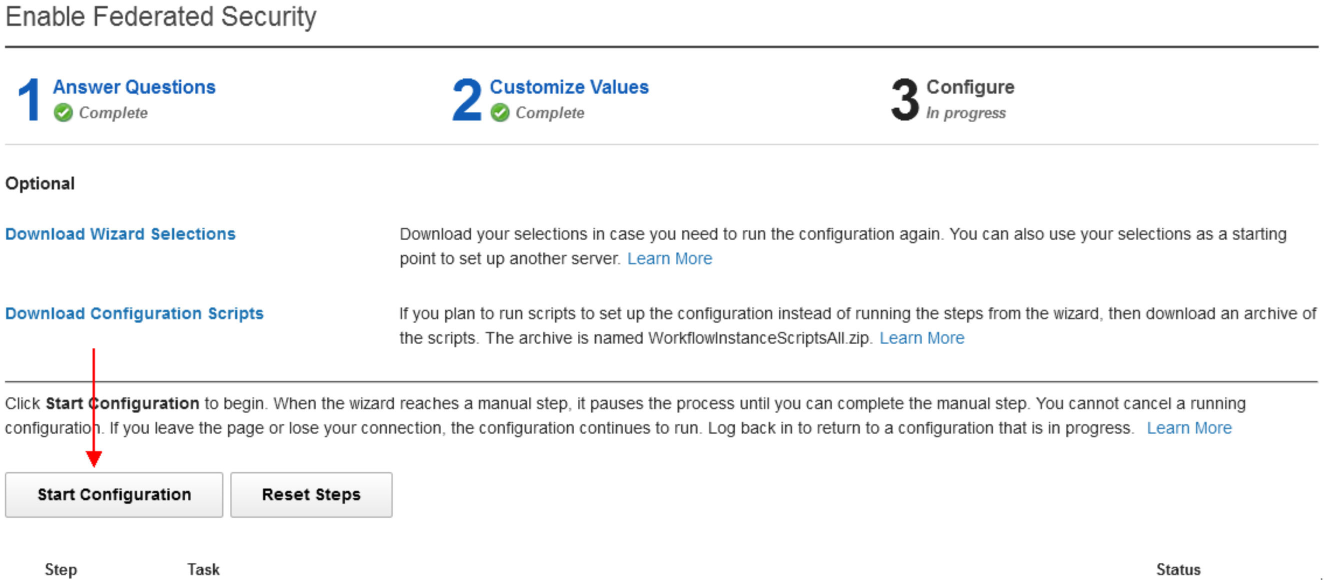
- The *default parent* configurations per-entity-type will be used as defaults. New users and groups will be created here if you use the default administrative portlets. If needed, the PUMA API allows programmers to specify the base DN for new users, thereby overriding these defaults.

10. Expand the advanced section and verify or update those additional configuration points:



*LDAP group objectclasses:	groupOfUniqueNames	Example: groupOfUniqueNames
LDAP group objectclasses for creating groups:		Example: groupOfUniqueNames;myPortalObjectClass
LDAP group search bases:		Example: "cn=groups1,dc=yourco,dc=com;cn=groups2,dc=yourco,dc=com"
*LDAP PersonAccount objectclasses:	inetOrgPerson	Example: inetOrgPerson
LDAP PersonAccount objectclasses for creating users:		Example: inetOrgPerson;myPortalObjectClass
LDAP search bases for the PersonAccount:		Example: "cn=users1,dc=yourco,dc=com;cn=users2,dc=yourco,dc=com"
Group dummy member:	uid=dummy	
*Group member attribute:	uniqueMember	Example: uniqueMember
*Group object class:	groupOfUniqueNames	Example: groupOfNames
*GM member attribute scope:	Direct	
Membership attribute name:	ibm-allGroups	Example: ibm-allGroups
GC member attribute scope:	Direct	
Certificate filter:		Example: uid=${SubjectCN}
Certificate map mode:	Exact DN	
Group RDN attribute:	cn	Example: cn
PersonAccount RDN attribute:	uid	Example: uid

11. Click **Start Configuration**:



Enable Federated Security

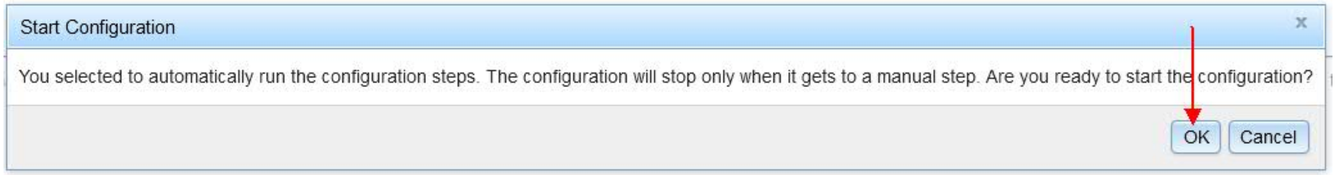
Optional

Download Wizard Selections — Download your selections in case you need to run the configuration again. You can also use your selections as a starting point to set up another server. Learn More

Download Configuration Scripts — If you plan to run scripts to set up the configuration instead of running the steps from the wizard, then download an archive of the scripts. The archive is named WorkflowInstanceScriptsAll.zip. Learn More

Click **Start Configuration** to begin. When the wizard reaches a manual step, it pauses the process until you can complete the manual step. You cannot cancel a running configuration. If you leave the page or lose your connection, the configuration continues to run. Log back in to return to a configuration that is in progress. Learn More

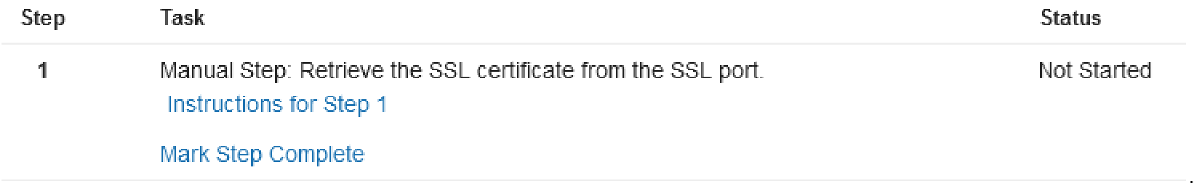
12. Click **OK**:



Start Configuration

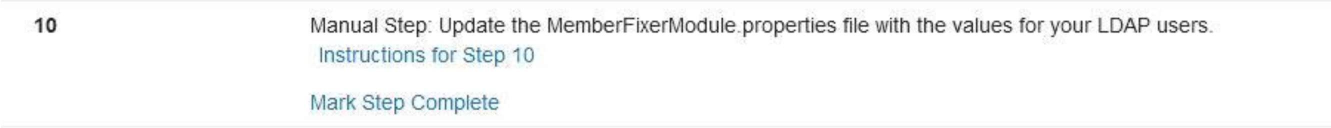
You selected to automatically run the configuration steps. The configuration will stop only when it gets to a manual step. Are you ready to start the configuration?

13. If you chose to enable SSL, the first step will be a manual step to retrieve the SSL certificate from the LDAP server's SSL port. Click on “Instructions for Step 1” and follow the instructions to retrieve the signer certificate and add it to the server's truststore. After the signer certificate is imported, click “Mark Step Complete”.



Step	Task	Status
1	Manual Step: Retrieve the SSL certificate from the SSL port. Instructions for Step 1 Mark Step Complete	Not Started

14. The Configuration Wizard may ask you to take manual steps to run WCM's Member Fixer. Click “**Instructions for step X**” and follow the detailed instructions:



10	Manual Step: Update the MemberFixerModule.properties file with the values for your LDAP users. Instructions for Step 10 Mark Step Complete

15. The last manual step is generally *Manual Step: Map attributes to ensure proper communication between WebSphere Portal and the LDAP server*. For this step, just click **Mark step complete** because attribute mapping is addressed more thoroughly in Task 11 [below].
16. Test:
    - Logging in with a sample user from the LDAP *using his **uid*** (for example, the user from Task 9, #5 [above]). Alternative or additional login properties are addressed in Task 11 (below).
    - Log in as the HCL Portal super-administrator. Go to wps/myportal/Administration > Manage Users and Groups.
        - Verify that you can search for the users and groups from the LDAP.
        - Verify that group membership resolves correctly, per user.
        - Verify that group members resolve correctly, per static group (not dynamic groups – see Task 11 [below]).
    - If these tests are successful, run **backupConfig** to back up the WebSphere Application Server configuration.

HCL Portal is now integrated with your LDAP for the major security functions of:

- Authentication – verifying users' identities (i.e. logging in with uid)
- Authorization – determining users' rights to access resources, most commonly managed based on group membership

Task 11 - Realms, User and Group Management, and Attribute Mapping

In this task, you will perform advanced configuration to meet the specific requirements of your application and environment.

1. If your application requires additional base entries per LDAP to implement realms for virtual portals (Task 4, #8 [above]) add those now with command lines like:
    - **./ConfigEngine.sh wp-create-base-entry -Did=*MyLDAP1* -DbaseDN=o=org2,c=us**

-DnameInRepository=o=org2,c=us

- Most commonly, each LDAP is associated with only one base entry, so this step is not generally required.
- Base entries should not overlap per realm.
- Add these base entries to your realm definitions according to the InfoCenter section entitled Adding realm support. See also Task 13 (below).

2. Validate the attribute configuration to check for any properties defined in VMM that do not have corresponding LDAP attributes:
    - **./ConfigEngine.sh wp-validate-federated-ldap-attribute-config**
    - Check ConfigTrace.log for:
    - *Possible problems for PersonAccount* and:
    - *Possible problems for Group*
    - Problems could occur if your application requested properties in this list.
3. Refer to Task 4, #9 (above) for which user and group attributes your application needs to access.
    - Are any of these in the *Possible problems* list from Task 11, #2 (above)? Specifically, in the list "*The following attribues are defined in Portal but not in LDAP*"? If so, map them to the appropriate attributes in LDAP, *per entity type*, with command lines like:
    - **./ConfigEngine.sh wp-update-federated-ldap-attribute-config -Dfederated.ldap.attributes.mapping.ldapName=mail -Dfederated.ldap.attributes.mapping.portalName=ibm-primaryEmail -Dfederated.ldap.attributes.mapping.entityTypes=PersonAccount**
    - Stop and restart the appropriate servers to synchronize (if clustered) and propagate the configuration changes. Refer to Starting and stopping servers, deployment managers, and node agents.
    - The mapping in the example above, *ibm-primaryEmail* (VMM property) mapping to *mail* (LDAP attribute), is a common requirement for HCL components like People Finder and WCM workflows that use email.
    - This task may alternatively read from properties files and may read in lists for these properties. This example uses a single attribute mapping for simplicity.
4. Add and optionally map all other attributes your application needs to access:
    - Run: **./ConfigEngine.sh wp-query-attribute-config**
    - Compare wp_profile_root/ConfigEngine/log/availableAttributes.html with the list from Task 4, #9 (above). Are any required attributes not available? If so, proceed with:
    - Prepare the system for adding attributes:
        - **./ConfigEngine.sh wp-la-install-ear**
    - Update wkplc.properties (or your helper file) with entries like:
        - la.providerURL=corbaloc:iiop:localhost:10031
            - In a cluster, the provider URL should point to the DMGR and its bootstrap port, like: la.providerURL=corbaloc:iiop:dmgr.hcl.com:9809
        - la.propertyName=customAttribute
        - la.entityTypes=PersonAccount,Group *(which entity types have this property)*
        - la.dataType=String
        - la.multiValued=false
    - Add the custom property with a command line like:
        - **ConfigEngine.sh wp-add-property**
    - Optionally map custom VMM properties to LDAP attributes as in Task 11, #3 (above). If not mapped, VMM uses the VMM property name as the LDAP attribute name in

requests to the LDAP.

5. If your application requires login by full DN, enable it now with:
   - **./ConfigEngine.sh wp-modify-realm-enable-dn-login**
6. If the tests in Task 9, #7 & #8 (above) failed, configure VMM to use the distinguished name as the external identifier.
7. If your application requires login attributes other than or in addition to *uid*:
   - Log in to the WAS ISC as the WAS administrator.
   - Go to **Global security > Federated repositories > MyLDAP1**
   - Update **Federated repository properties for login** as needed. Refer to WAS ISC help panels. For example: *uid;loginAttr2*
   - Save the configuration, synchronize (if clustered), and restart server(s) to load the updated configuration.

Task 12 - Other Performance, LDAP, and Networking Considerations

1. To improve HCL Portal performance, you may disable nested groups if your application does not require HCL Portal to resolve them. Consider this especially if nested groups are resolved by an LDAP membership attribute. Refer to your application requirements and test results in Task 3, Part A, #12 (above); Task 4, #4 (above); Task 8, Part B, #2 & 3 (above); and Task 8, Part C, #5 (above). To disable HCL Portal resolving nested group membership:
   - Follow the steps in the HCL Product Documentation to disable nested groups.
2. WebSphere Application Server resolves group membership when users log in. With a federated repository, VMM handles both authentication and resolving group membership. Evaluate how group membership is managed in your environment and consider configuring HCL Portal to reuse the group membership as determined by WebSphere Application Server to improve performance.
   - This is most common if a group membership attribute has a scope of *nested* (and the application does not use dynamic groups) or *all* (and the application uses dynamic groups).
   - If your application requires HCL Portal to resolve nested groups (see Task 12, #1 [above]), then you should not configure group reuse.
3. You must manually enable dynamic groups if your application requires them and if the membership attribute does not resolve them (Task 3, Part A, #13 [above]):
   - Log in to the WebSphere Application Server Integrated Solutions Console
   - Select **Security > Global Security**.
   - Under **User account repository**, select **Federated repositories** for the **Current realm definition** and click **Configure**.
   - Under **Related Items**, click **Manage repositories**.
   - Click on ***MyLDAP1***.
   - Under **Additional Properties**, click **Group attribute definition** then click **Dynamic member attributes**.
   - Click **New** and specify values for the **Name** and **Object class** fields per Task 3, Part A, #11 (above) and as verified in Task 8, Part D (above). For example,
     - **Name**: *memberurl*
     - **Object class**: *groupofurls*
4. You *may* manually disable referrals if your application does not require it (Task 3, Part A,

#15 [above]). Recall that the LDAP browser verification steps did not use referrals, unless specifically configured.

- If you configure VMM to follow LDAP referrals, VMM will try to connect to the referred-to server with the bind user and password. If the referred-to LDAP cannot authenticate this bind user, referrals will fail, as will any VMM/WAS/PUMA functions that rely upon any such referral. Verify that any other LDAP servers that the primary LDAP server may refer to can authenticate the bind user with steps like those in Task 5 (above).

To disable LDAP referrals:

- Log in to the WebSphere Application Server Integrated Solutions Console
- Select **Security > Global Security**.
- Under **User account repository**, select **Federated repositories** for the **Current realm definition** and click **Configure**.
- Under **Related Items**, click **Manage repositories**.
- Click on ***MyLDAP1***.
- Under **Support referrals to other LDAP servers**, select **ignore**.
- Click **OK** and **Save**.

5. Configure LDAP failover, if needed (Task 3, Part A, #8 [above]). Refer to IBM Knowledge Center.
6. If you chose for your administrative user and group to reside in LDAP, then you *may* remove the default file repository. Doing so helps avoid conflicts between user IDs in the file repository (generally used for testing) and users in the LDAP (generally live users). Remove the file repository from the federated repository with a command line like:
    - **./ConfigEngine.sh wp-delete-repository -Dfederated.delete.baseentry=o=defaultWIMFileBasedRealm -Dfederated.delete.id=InternalFileRepository**
7. If required by your application, add search filters per-entity-type.
    - The searchFilter properties let you override the default search filters that VMM uses to distinguish entity types. In general, this is not required because VMM constructs the search filter based on the object class of the entity type. If you choose to specify a searchFilter based on Task 4, #6 (above), be sure to encode it correctly and include the object class.
8. By default, VMM maintains a pool of connections to the LDAP server to avoid the overhead of establishing TCP connections for each request. VMM refers to this as the *context pool*. This pool has the added benefit of limiting the total number of connections and requests VMM makes to LDAP at any given time. Proxies, such as load balancers and certain firewalls can interfere with the way this pool functions, affecting performance. Configure the context pool according to the requirements of your application and network (Task 3, Part A, #7 [above]).
    - Log on to the WebSphere Application Server Integrated Solutions Console and navigate to **Global security > Federated repositories > Manage repositories > MyLDAP1 > Performance**.
    - Check **Context pool times out**
    - Set the value in **Seconds** according to your requirements. You should ensure that VMM does not use any LDAP connections that might be older than any idle timeouts imposed by proxies.
    - If you do not have complete information about firewalls, proxies, etc. use 2700. With this setting, VMM uses the context pool for performance, but avoids default TCP timeouts on most proxies.

- Refer to these articles for additional guidance:
  - Limiting LDAP Connections
  - Timing connections out of the context pool before they go stale

Task 13 - Consider other requirements, beyond the scope of this guide

1. Your application may need to immediately load certain time-sensitive data from the LDAP on demand, rather than via intermediate caches (generally designed to reduce load on LDAP servers and improve application performance). If so, refer to this article for guidance.
2. Your application may need to store information about users or groups independently of LDAP (Task 4, #10 [above]). If so, implement property extension per the InfoCenter.
3. If your application needs to create or manage static groups, but the LDAP bind user does not have sufficient access rights to manage these in the LDAP, then enable application groups.
4. If your application needs to create and manage an equivalent to dynamic groups, consider configuring the rule-based user groups adapter. Rule-based groups function much like dynamic groups. Rule-based groups are stored in a database repository.
5. HCL Web Content Manager may impose additional requirements on your LDAP integration. Refer to the HCL Product Documentation for specific steps, such as those for:
   - Running Member Fixer to update ownership of components in the Internet and Intranet Site Templates.
   - Assigning access to custom web content libraries.
   - Running Member Fixer against custom web content libraries you may have created prior to LDAP integration.
6. Consider your requirements for virtual portals (Task 4, #8 [above]). Refer to the HCL Product Documentation for specific instructions on:
   - Creating realms
   - Adding base entries to realms
   - Modifying entity type configurations per realm

Resources

- HCL Portal Product Documentation
- Security Hardening Guide for HCL Digital Experience
- HCL Portal Support Site
- Internet Engineering Task Force (various RFCs)
- Internet Assigned Numbers Authority