HITACHI
Inspire the Next

# Configuring Pentaho with LDAP or Active Directory

Change log (if you want to use it):

| Date | Version | Author | Changes |
|---|---|---|---|
| 07/2018 | 1.0 | Carlos Lopez | |
| 02/2020 | 1.1 | Carlos Lopez | Change to `populator.groupSearchFilter` line in Populator section |
| | | | |

# Contents

# Overview

Pentaho can be configured to use multiple mechanisms for authentication and authorization, including Microsoft Active Directory (MSAD), OpenLDAP, or database-based authentication, also known as Java Database Connectivity (JDBC) authentication.

This document works through the steps needed to set up Pentaho to authenticate using MSAD, and explains the components used in the `applicationContext-security-ldap.properties`.

The intention of this document is to speak about topics generally; however, these are the specific versions covered here:

| Software | Version(s) |
|---|---|
| Pentaho | 7.x, 8.x, 9.0 |

The Components Reference in Pentaho Documentation has a complete list of supported software and hardware.

## Before You Begin

Before beginning, use the following information to prepare for the procedures described in the main section of the document.

### *Prerequisites*

This document assumes that you have the following:

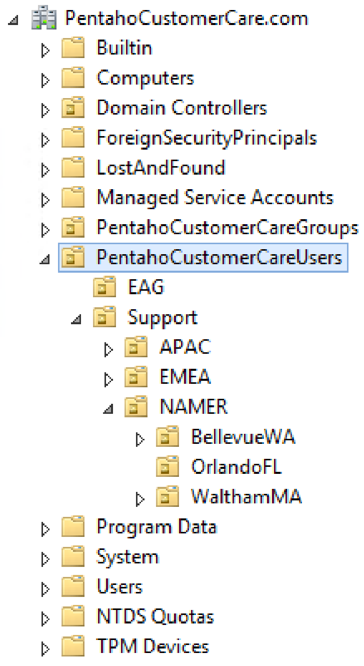
- MSAD installed on a MS Windows Server 2012 R2
- Pentaho 7.x+ installed and ready to be configured
- A binding account that will be used to connect from Pentaho to MSAD; this user does not have to be an administrator
- An administrator account that is a system administrator for Pentaho

### *Configuration for Document Examples*

Ideally, you would use Pentaho-related Active Directory groups, but this will depend on your configuration.

- Domain: `PentahoCustomerCare.com`
- Organizational Unit (OU): `PentahoCustomerCareGroups`
- Organizational Unit: `PentahoCustomerCareUsers`

Consider our scenario to connect to LDAP/Active Directory with the following structure:



```
PentahoCustomerCare.com
    Builtin
    Computers
    Domain Controllers
    ForeignSecurityPrincipals
    LostAndFound
    Managed Service Accounts
    PentahoCustomerCareGroups
    PentahoCustomerCareUsers
        EAG
        Support
            APAC
            EMEA
            NAMER
                BellevueWA
                OrlandoFL
                WalthamMA
    Program Data
    System
    Users
    NTDS Quotas
    TPM Devices
```

*Figure 1: Scenario Structure*

## *Authentication and Authorization*

To learn how to configure Pentaho to use an authentication scheme, you first need to understand what authentication and authorization are in terms related to Pentaho. For authentication and authorization purposes, Pentaho uses Spring Framework.

**Authentication** happens when the user logs in. The system checks to see:

- Whether the user is a valid user to log in
- Whether the user is active

Once the user is proven to be active, and can log in, **authorization** occurs where we check the roles the user belongs to.

**Roles**:

- Represent what the user is authorized to do in the server
- Are assigned only once we know who the user is
- Give the user operational permissions on the server such as **Manage Security**, **Schedule Content**, **Manage Data Sources**, and others.

*Note that a user may be able to open a report, but still may not be allowed to see its contents; this is not to be confused with authorization. Being able to see the contents of a report is controlled through Mondrian roles in Analyzer reports. These are security-constrained accesses and are beyond the scope of this document.*

# Connecting and Binding to LDAP/Active Directory Server

The first step to configure Pentaho with Active Directory is to connect and bind to your LDAP/Active Directory server.

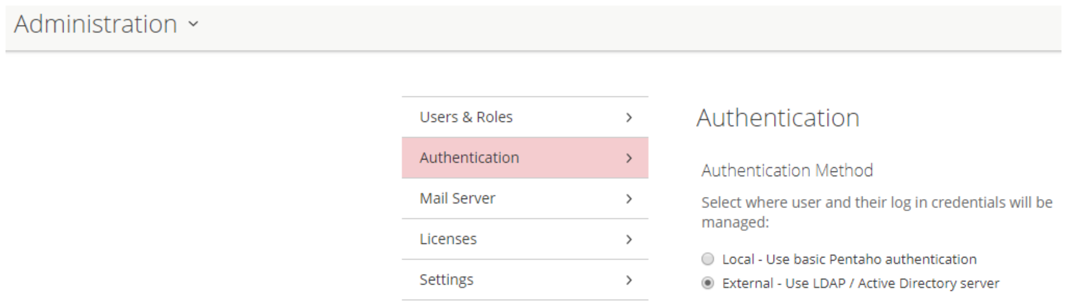
You can find details on these topics in the following sections:

- Connecting to Active Directory Server
- Securing LDAP Password
- Binding to LDAP/Active Directory Server
- URL and Port Considerations
- Settings File Location

## Connecting to LDAP/Active Directory Server

To connect to your server:

1. In the **Pentaho User Console (PUC)**, choose the **Administration** perspective.
2. Click on **Authentication**.
3. Choose **External – Use LDAP/Active Directory server**.



Authentication Method

Select where user and their log in credentials will be managed:

- Local - Use basic Pentaho authentication
- [✓ External - Use LDAP / Active Directory server]

*Figure 2: Changing Authentication Method in PUC*

## Encoding LDAP Password

Pentaho provides a service (`IPasswordService`) that allows the encryption and decryption of strings that have `Base64` as the default encoding/decoding scheme. Other schemes, such as AES or Triple DES, can be implemented.

You can use a Spring Expression Language (SpEL) query to access this service and use it to decode a string from a properties file, then assign it to the Spring variable that holds this password.

In this example, we will use Base64 encoding. To use a different encoding/decoding scheme you will need to implement the IPasswordService with your desired method:

1. Stop the Pentaho Server.
2. Run your password through a Base64 encoder. An example password is Password1, which results in an encoded password of UGFzc3dvcmQx.
3. Open the pentaho-solutions/system/applicationContext-security-ldap.properties file with any text editor.
4. Edit to assign the encoded value to the contextSource.password property, then save and close the file:

```
contextSource.password=UGFzc3dvcmQx
```

5. Open the pentaho-solutions/system/applicationContext-spring-security-ldap.xml.
6. Change the password property value to use the SpEL query as shown below:

```
<bean id="contextSource"
class="org.springframework.security.ldap.DefaultSpringSecurityContextSource
">
    <constructor-arg value="${ldap.contextSource.providerUrl}"/>
    <property name="userDn" value="${ldap.contextSource.userDn}"/>
    <property name="password"
value="#{IPasswordService.decrypt('${ldap.contextSource.password}')}"/>
  </bean>
```

7. Save and close the file.
8. Start the Pentaho Server.

## Binding to LDAP/Active Directory Server

To bind the LDAP/Active Directory server, under **LDAP Server Connection**, populate the following entries:

1. **Server URL**: Enter your server's DNS name or IP address in this format: ldap://server_name_or_ip_address:port_number. In our example, we will use this IP address and port 389: ldap://10.100.7.17:389.
2. **Username**: This is the binding account. It does not have to be an LDAP/Active Directory administrator, but only needs read access.
3. **Password**: Enter the correct password for the username.

## URL and Port Considerations

You may need to use a different port, depending on the type of configuration your Active Directory is set to. If you are using a global catalog, consider using port 3268. If you are using SSL, consider using LDAPS and port 636. The table below is a good reference:

*Table 1: Protocols and Ports*

| Protocol | Port | AD and AD Domain Services Usage | Type of Traffic |
|---|---|---|---|
| TCP and UDP | 389 | Directory, Replication, User and Computer Authentication, Group Policy, Trusts | LDAP |
| TCP | 636 | Directory, Replication, User and Computer Authentication, Group Policy, Trusts | LDAP SSL |
| TCP | 3268 | Directory, Replication, User and Computer Authentication, Group Policy, Trusts | LDAP GC |
| TCP | 3269 | Directory, Replication, User and Computer Authentication, Group Policy, Trusts | LDAP GC SSL |
| TCP and UDP | 88 | User and Computer Authentication, Forest Level Trusts | Kerberos |
| TCP and UDP | 53 | User and Computer Authentication, Name Resolution, Trusts | DNS |

You can find more information in Microsoft Technet for Active Directory Domain Services port requirements.

## Settings File Location

The settings that you put in for the authentication method are saved in the file `/pentaho-server/pentaho-solutions/system/applicationContext-security-ldap.properties`, in the following lines:

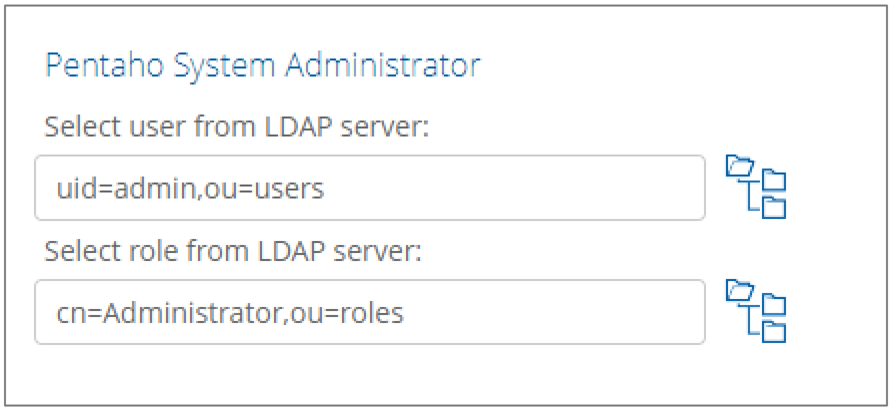
```
contextSource.providerUrl=ldap\://10.100.7.17\:389


contextSource.password=Password1


contextSource.userDn=CN\=Elena
Neill,OU\=OrlandoFL,OU\=NAMER,OU\=Support,OU\=PentahoCustomerCareUsers,DC\=
PentahoCustomerCare,DC\=com
```

# Configuring the Pentaho System Administrator

In this section, you tell the Pentaho server which LDAP/Active Directory user and group it should consider to be Pentaho's admin user and administrator role. You can find details on these topics in the following sections:

- Changing Username and Role
- Considerations for the Pentaho System Administrator Section

The user is `admin` with the role `Administrator` in Pentaho's default security.



Pentaho System Administrator

Select user from LDAP server:

uid=admin,ou=users

Select role from LDAP server:

cn=Administrator,ou=roles

*Figure 3: Default Username and Role for Administrator*

## Changing Username and Role

In this case, we are going to replace the default username and role with the following:

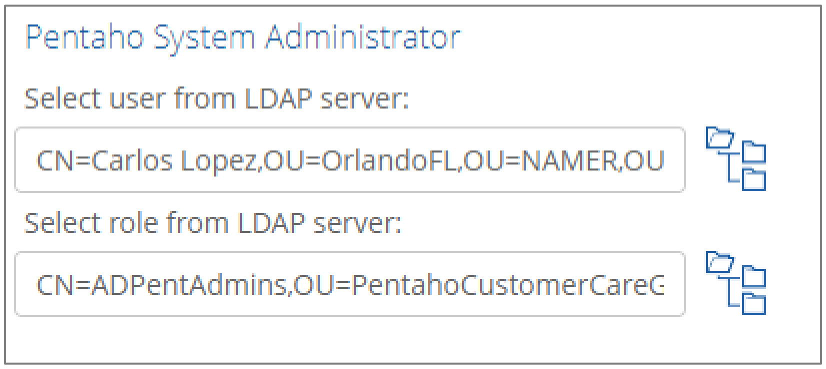
1. Select user from the LDAP server:

```
CN=Carlos
Lopez,OU=OrlandoFL,OU=NAMER,OU=Support,OU=PentahoCustomerCareUsers,DC=Penta
hoCustomerCare,DC=com
```

2. Select role from the LDAP server:

```
CN=ADPentAdmins,OU=PentahoCustomerCareGroups,DC=PentahoCustomerCare,DC=com
```

3. Your screen should now look like this:


Pentaho System Administrator

Select user from LDAP server:

CN=Carlos Lopez,OU=OrlandoFL,OU=NAMER,OU

Select role from LDAP server:

CN=ADPentAdmins,OU=PentahoCustomerCareG

*Figure 4: New Username and Role for Administrator*

## Considerations for the Pentaho System Administrator Section

The user being identified here is meant to replace the `admin` user from the default Jackrabbit configuration. This user must:

- Be a member of the same group that will be considered the `Administrator` group in Pentaho (any user in that group can be an administrator, not just one user). In our example, this means we can use any member of the group `ADPentahoAdmins`.
- Always be active, as the user is used by Pentaho to do queries and tasks related to user loading and role mapping between Pentaho and your LDAP/Active Directory server.

Consider using the full Distinguished Name (DN) for both user and role.

These settings are saved in the file `/pentaho-server/pentaho-solutions/system/applicationContext-security-ldap.properties` in the following lines:

```
adminUser=CN\=Carlos
Lopez,OU\=OrlandoFL,OU\=NAMER,OU\=Support,OU\=PentahoCustomerCareUsers,DC\=
PentahoCustomerCare,DC\=com

adminRole=CN\=ADPentAdmins,OU\=PentahoCustomerCareGroups,DC\=PentahoCustome
rCare,DC\=com
```

# Configuration

The next step will be to set up the configuration of the following items:

- Choosing LDAP Provider
- User Search
- Roles
- Populator
- Wrapping Up Configuration

## Choosing LDAP Provider

Next, configure the LDAP provider by choosing **Custom Configuration** from the **LDAP Provider** dropdown:

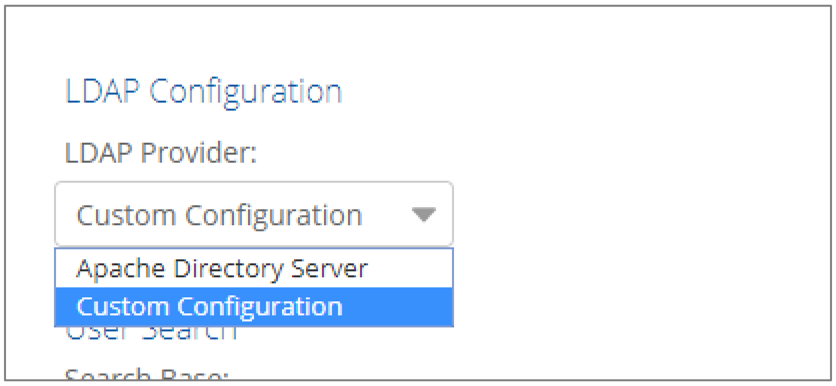


*Figure 5: LDAP Provider*

These settings are saved as `providerType=ldapCustomConfiguration` in the file `/pentaho-server/pentaho-solutions/system/applicationContext-security-ldap.properties`.

## User Search

Configuring the **User Search** is an important piece of the process, because it is what controls which users from your LDAP/Active Directory server will be able to log into Pentaho.

1. The **Search Base** box tells Pentaho which OU contains the users who will be able to log into PUC. Any groups/users under this OU will have access to log into Pentaho. In this example, we will use `PentahoCustomerCareUsers`:

```
OU=PentahoCustomerCareUsers,DC=PentahoCustomerCare,DC=com
```

2. The **Search Filter** box tells Pentaho which AD attribute will be used for login, either `sAMAccountName` (where the user would type in just their username, such as `clopez`) or

`UserPrincipalName` (where the user would type in their username and domain, such as `clopez@pentahocustomercare.com`). In our example, we use `sAMAccountName`:

```
(sAMAccountName={0})
```

*We recommend you consider using the AD attribute that matches the one your users type in when they log into your AD network.*

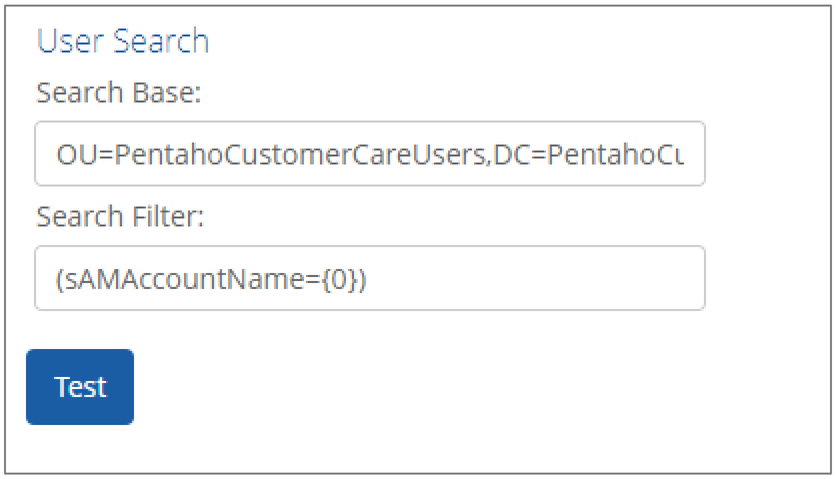


*Figure 6: User Search*

These settings are in `/pentaho-server/pentaho-solutions/system/applicationContext-security-ldap.properties` in the following lines:

```
userSearch.searchBase=OU\=PentahoCustomerCareUsers,DC\=PentahoCustomerCare,
DC\=com


userSearch.searchFilter=(sAMAccountName\={0})
```

## Considerations for the Search Filter

The filter allows you to have a more granular control of who can access PUC. In our example, we configured Pentaho to do a general search for the users under `PentahoCustomerCareUsers` and to use the `sAMAccountName` attribute from every user to access PUC.

As a different option, in this next code, we are telling Pentaho to grant access to those users that are members of the `ADPPentCustomerCare` Group below the `PentahoCustomerCareUsers` OU:

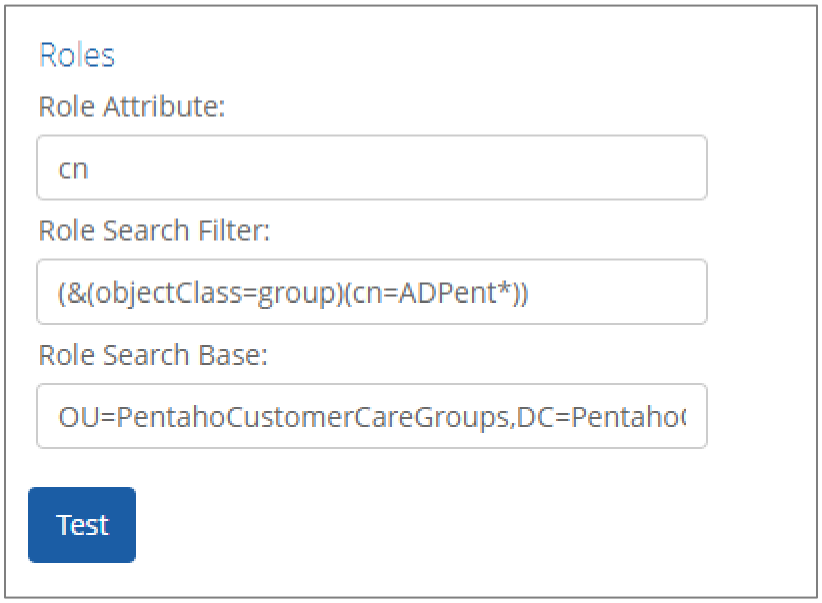
```
userSearch.searchBase=OU\=PentahoCustomerCareUsers,DC\=PentahoCustomerCare,
DC\=com


userSearch.searchFilter=(&(sAMAccountName\={0})(memberOf\=CN\=ADPentCustome
rCare,OU\=PentahoCustomerCareGroups,DC\=PentahoCustomerCare,DC\=com))
```

## Roles

The **Roles** settings tell Pentaho where to search for the roles that are going to be displayed and used through the PUC.

These settings are in `/pentaho-server/pentaho-solutions/system/applicationContext-security-ldap.properties` in the following lines:

```
allAuthoritiesSearch.roleAttribute=cn

allAuthoritiesSearch.searchBase=OU\=PentahoCustomerCareGroups,DC\=PentahoCu
stomerCare,DC\=com

allAuthoritiesSearch.searchFilter=(&(objectClass\=group)(cn\=ADPent*))
```


Roles

Role Attribute: cn

Role Search Filter: (&(objectClass=group)(cn=ADPent*))

Role Search Base: OU=PentahoCustomerCareGroups,DC=Pentaho(

*Figure 7: Roles*

1. The **Role Attribute** box tells Pentaho what to display when the roles are populated. In this example, we will use the common name (CN) of the group. Groups, Roles, and Authorities are used interchangeably.
2. The **Role Search Filter** box tells Pentaho what to search for. In our example, we will use the `objectClass` attribute of the group:

   ```
   (objectClass=group)
   ```

   Note that in our example, we are filtering down not to just groups, but to those groups whose common name starts with `ADPent*`. By using that wildcard, we can include groups such as `ADPentAdmins`, `ADPentUsers`, and `ADPentEAG`:

   ```
   (&(objectClass=group)(cn=ADPent*))
   ```

3. The **Role Search Base** box tells Pentaho where to start searching for the groups that have access to Pentaho. In our example, we are using these attributes:

```
OU=PentahoCustomerCareGroups,DC=PentahoCustomerCare,DC=com
```

*When you are configuring these roles settings, keep in mind that these settings are used in multiple places in PUC. Depending on the number of roles and groups you have, they can take some time to load.*

4. Under Administration, click on **Users & Roles**.
5. Click on **Manage Roles**. You can configure specific operation permissions for roles from here:

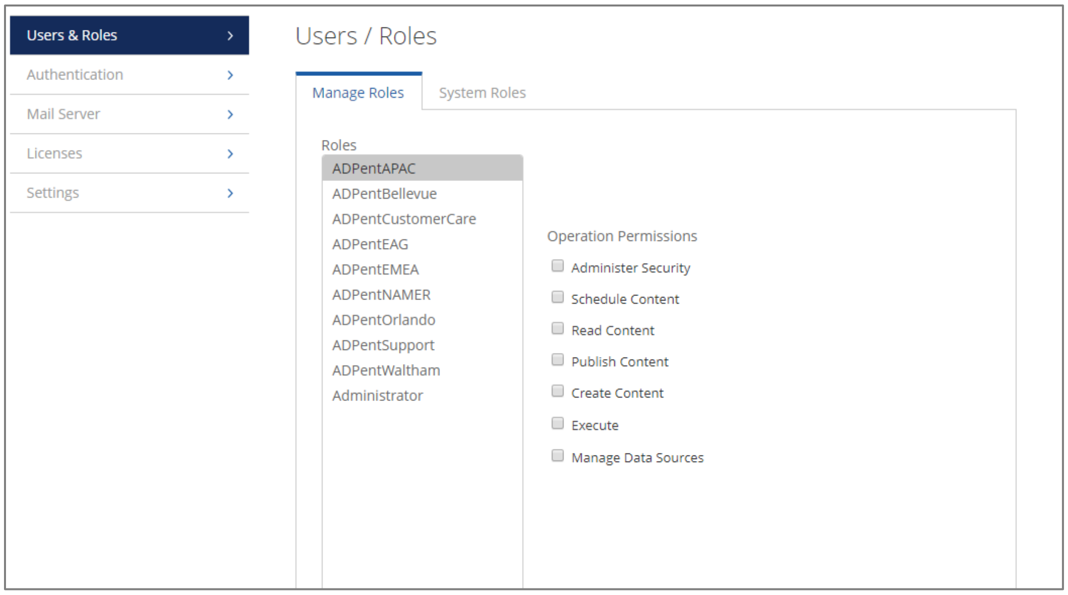
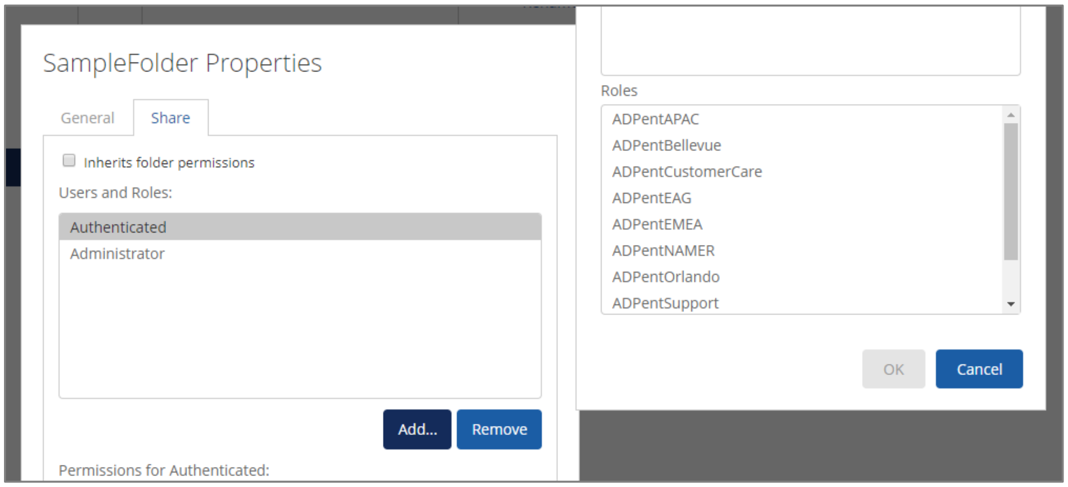


*Figure 8: Users & Roles*

6. Add or remove role permissions for specific folders by going to **Browse Files**, highlighting a file in the middle pane, and then clicking **Share…** under the **File Actions** pane. Under the **Share** tab under **Roles**:


SampleFolder Properties

General | Share

☐ Inherits folder permissions

Users and Roles:

Authenticated
Administrator

Add... | Remove

Permissions for Authenticated:

Roles

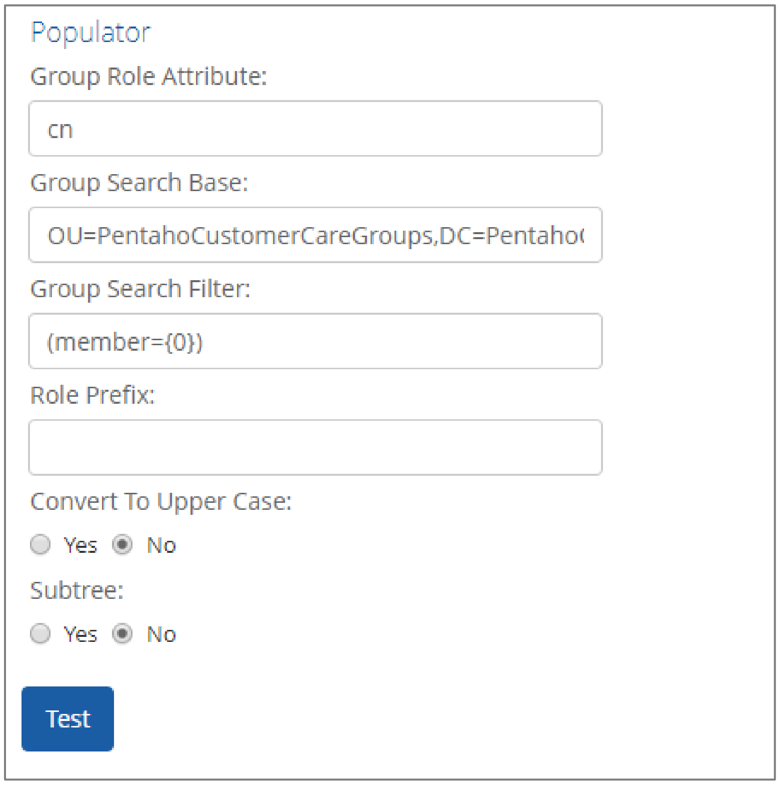
ADPentAPAC
ADPentBellevue
ADPentCustomerCare
ADPentEAG
ADPentEMEA
ADPentNAMER
ADPentOrlando
ADPentSupport

OK | Cancel

*Figure 9: Folder Properties*

## Populator

This section populates or gathers the groups a user is a member of during the active session.


Populator

Group Role Attribute: cn

Group Search Base: OU=PentahoCustomerCareGroups,DC=Pentaho

Group Search Filter: (member={0})

Role Prefix:

Convert To Upper Case: Yes [✓ No]

Subtree: Yes [✓ No]

Test

*Figure 10: Populator*

1. The **Group Role Attribute** box tells Pentaho what to display in the login session as the display attribute. In this example, we will use the CN.
2. The **Group Search Base** box tells Pentaho where the AD groups will be located for the session. In this example:

```
OU=PentahoCustomerCareGroups,DC=PentahoCustomerCare,DC=com
```

3. The **Group Search Filter** box tells Pentaho what attribute to use from the AD groups. In this example:

```
(member={0})
```

4. **Role Prefix**, **Convert to Upper Case**, and **Subtree** are legacy attributes and do not need configuration.

These settings are in `/pentaho-server/pentaho-solutions/system/applicationContext-security-ldap.properties` in the following lines:

```
populator.groupRoleAttribute=cn

populator.groupSearchBase=OU\=PentahoCustomerCareGroups,DC\=PentahoCustomer
Care,DC\=com

populator.convertToUpperCase=false

populator.searchSubtree=false

populator.groupSearchFilter=(&(member\={0})(objectClass\=group)(cn\=ADPent*
))

populator.rolePrefix=
```

### *Considerations when Configuring the Populator*

Things to consider when you are configuring the populator include:

- Use the same Group Search Base as the one in the Roles Group Search Base, as this will eliminate performance issues when the user is accessing PUC.
- Use a filter to remove those users who are `active` or `enabled` *only*; see this example and change it accordingly:

```
(&(memberof\=CN\=ADPentCustomerCare,OU\=PentahoCustomerCareGroups,DC\=Penta
hoCustomerCare,DC\=com)(!(userAccountControl:1.2.840.113556.1.4.803:=2)))
```

## Wrapping Up Configuration

You have now completed all the configurations needed to connect to your LDAP/Active Directory Server. Next:

1. Click **Save**.
2. Restart the Pentaho Server.

3. The following files will be changed:
    a. `pentaho-server/pentaho-solutions/system/security.properties`, where the provider will change from `provider=jackrabbit` to `provider=ldap`.
    b. `pentaho-server/pentaho-solutions/system/repository.properties`, where `singleTenantAdminUserName=admin` will change to `singleTenantAdminUserName=clopez` (the administrator account used as the Pentaho System Administrator).
4. The default configuration will be changed to the one saved by the Authentication Perspective on PUC.

# Manual Configurations

For more options, you can manually configure other properties. Further information is available in the Pentaho wiki at Enabling Verbose LDAP Logging, LDAP Troubleshooting, Nested Roles or Groups (LDAP), and Using Active Directory.

More information in this document is available on these topics:

- Encoding LDAP Password
- Configuring Subtree for Roles
- Usernames on the Share Tab
- Configuring Subtree for Users
- Configuring Admin User for `pentaho.jms.cfg`

## Encoding LDAP Password

By default, the `org.pentaho.platform.util.KettlePasswordService` class is used by `IPasswordService` to decode Base64 encoded strings or strings encoded with the Kettle utility, `Encr.bat` (or `encr.sh` for Linux).

This subsection covers which files to edit and what Java class is used to decode passwords. It assumes that you are familiar with the `applicationContext-security-ldap.properties` file and that you have a working Pentaho Server already configured with LDAP or Active Directory authentication. More information about related topics outside of this document can be found at:

- Switching to MS Active Directory
- Switching to LDAP
- Manual LDAP Configuration

### *Configure Pentaho Server for Encoded LDAP or Active Directory Passwords*

This section will guide you through configuring your server to use these encoded passwords. Follow these steps using a text editor:

1. Open the `applicationContext-spring-security-ldap.xml` file located in `pentaho-server/pentaho-solutions/system`.
2. Locate and comment out the following `contextSource` bean:

```
<bean id="contextSource"
class="org.springframework.security.ldap.DefaultSpringSecurityContextSource
">
    <constructor-arg value="${ldap.contextSource.providerUrl}"/>
    <property name="userDn" value="${ldap.contextSource.userDn}"/>
    <property name="password" value="${ldap.contextSource.password}"/>
</bean>
```

3. Add the following edited `contextSource` bean:

```
  <bean id="contextSource"
class="org.springframework.security.ldap.DefaultSpringSecurityContextSource
">
    <constructor-arg value="${ldap.contextSource.providerUrl}"/>
    <property name="userDn" value="${ldap.contextSource.userDn}"/>
    <property name="password"
value="#{IPasswordService.decrypt('${ldap.contextSource.password}')}"/>
  </bean>
```

Notice the change in the password value.

The Spring Expression Language has used `IPasswordService` to decrypt the string in the `applicationContext-securityldap.properties` file.

## Configure an Encoded Password with Base64

Base64 contains similar binary-to-text encoding schemes representing binary data in an ASCII string format by translating it into a radix-64 representation. This section covers steps for configuring an encoded password with Base64.

1. In a text editor, open the `applicationContext-security-ldap.properties` file located in `pentaho-server/pentaho-solutions/system`.
2. Edit the `contextSource.password` property with the encoded password.

For the Base64 encoded string `password123`, it would look like this:

```
contextSource.password=cGFzc3dvcmQxMjM\=
```

*If your encoded string contains an equals sign, like that shown in the encoded string above, you will need to add a backslash in front of it to escape it properly. Failure to do this will result in a failed authentication.*

## Configure an Encoded Password Using the Kettle Encr Script

If you have Pentaho Data Integration (PDI) client (Spoon) installed, you can use the `Encr.bat` (`encr.sh` for Linux) script to obfuscate a password for use in the `applicationContext-security-ldap.properties` file. This script is found in the root directory of Spoon where `Spoon.bat` (`spoon.sh` for Linux) is located.

1. To create an obfuscated password for the string `password123`, run the following in a command-line:

```
Encr.bat -kettle password123
```

You will get the following result after the script runs:

```
Encrypted 2be98afc86ad79397b80ea162dac3fd89
```

2. In a text editor, open the `applicationContext-security-ldap.properties` file located in `pentaho-server/pentaho-solutions/system`.
3. Edit the `contextSource.password` property with the obfuscated password from step 1. It should look like this:

```
contextSource.password=Encrypted 2be98afc86ad79397b80ea162dac3fd89
```

## Configuring Subtree for Roles

It is possible to configure subtree to search all the roles within the parent OU. Doing this tells Pentaho to search for all the folders within folders from the Group/Populator search base.

1. Locate the file `pentaho-server/pentaho-solutions/system/applicationContext-pentaho-security-ldap.xml`.
2. Locate the bean id `allAuthoritiesSearch`:

```
<bean id="allAuthoritiesSearch"
class="org.pentaho.platform.plugin.services.security.userrole.ldap.search.G
enericLdapSearch">
<constructor-arg index="0" ref="contextSource" />
<constructor-arg index="1">
<bean
class="org.pentaho.platform.plugin.services.security.userrole.ldap.search.L
dapSearchParamsFactoryImpl">
<constructor-arg index="0" value="${ldap.allAuthoritiesSearch.searchBase}"
/>
<constructor-arg index="1"
value="${ldap.allAuthoritiesSearch.searchFilter}" />
</bean>
</constructor-arg>
<constructor-arg index="2">
<bean
class="org.apache.commons.collections.functors.ChainedTransformer">
<constructor-arg index="0">
<list>
<bean
class="org.pentaho.platform.plugin.services.security.userrole.ldap.transfor
m.SearchResultToAttrValueList">
<constructor-arg index="0"
value="${ldap.allAuthoritiesSearch.roleAttribute}" />
</bean>
<bean
class="org.pentaho.platform.plugin.services.security.userrole.ldap.transfor
m.StringToGrantedAuthority">
<property name="rolePrefix" value="${ldap.populator.rolePrefix}" />
<property name="convertToUpperCase"
value="${ldap.populator.convertToUpperCase}" />
</bean>
</list>
</constructor-arg>
```

```
</bean>
</constructor-arg>
</bean>
```

3. Replace it with this bean:

```
<bean id="allAuthoritiesSearch"
      class="org.pentaho.platform.plugin.services.security.userrole.ldap.se
arch.GenericLdapSearch">
<constructor-arg index="0" ref="contextSource" />
<constructor-arg index="1">
<bean
class="org.pentaho.platform.plugin.services.security.userrole.ldap.search.L
dapSearchParamsFactoryImpl">
<constructor-arg index="0" value="${ldap.allAuthoritiesSearch.searchBase}"
/>
<constructor-arg index="1"
value="${ldap.allAuthoritiesSearch.searchFilter}" />
                         <constructor-arg index="2">
<bean class="javax.naming.directory.SearchControls">
<!-- 2 comes from
http://java.sun.com/javase/6/docs/api/javax/naming/directory/SearchControls
.html#SUBTREE_SCOPE -->
<property name="searchScope" value="2" />
</bean>
</constructor-arg>
</bean>
</constructor-arg>
<constructor-arg index="2">
<bean class="org.apache.commons.collections.functors.ChainedTransformer">
<constructor-arg index="0">
<list>
<bean
class="org.pentaho.platform.plugin.services.security.userrole.ldap.transfor
m.SearchResultToAttrValueList">
<constructor-arg index="0"
value="${ldap.allAuthoritiesSearch.roleAttribute}" /></bean>
<bean
class="org.pentaho.platform.plugin.services.security.userrole.ldap.transfor
m.StringToGrantedAuthority">
<property name="rolePrefix" value="${ldap.populator.rolePrefix}" />
<property name="convertToUpperCase"
value="${ldap.populator.convertToUpperCase}" />
</bean>
</list>
</constructor-arg>
</bean>
</constructor-arg>
</bean>
```

## Usernames on the Share Tab

To assign permissions to any object on PUC, you can use the Share tab to share permissions by users or roles.

*We do not recommend using the Share tab in a large environment, because the thousands of user accounts can cause performance issues. Instead, configure and manage users through roles, groups, or authorities.*

Usernames on the Share tab are not configured using the Authentication Perspective. Instead, they are configured in `\pentaho-server\pentaho-solutions\system\applicationContext-security-ldap.properties`.

Within these properties:

```
allUsernamesSearch.searchFilter=objectClass\=Person

allUsernamesSearch.searchBase=OU\=PentahoCustomerCareUsers,DC\=PentahoCusto
merCare,DC\=com

allUsernamesSearch.usernameAttribute=sAMAccountName
```

These properties are copied from the `allAuthoritiesSearch` attributes, but can be changed to match your environment.

*Use these instructions carefully, because you could end up with your entire Active Directory list of users in your search box, which could slow you down when you try to search for specific people.*

## Configuring Subtree

You can configure subtree to search all the roles within the parent OU.

Follow these instructions to show all the users on the Share tab:

1. Locate the file `pentaho-server/pentaho-solutions/system/applicationContext-pentaho-security-ldap.xml`.
2. Locate the bean `allUsernamesSearch`:

```
<bean id="allUsernamesSearch"
class="org.pentaho.platform.plugin.services.security.userrole.ldap.search.G
enericLdapSearch">
<constructor-arg index="0" ref="contextSource" />
<constructor-arg index="1">
<bean
class="org.pentaho.platform.plugin.services.security.userrole.ldap.search.L
dapSearchParamsFactoryImpl">
<constructor-arg index="0" value="${ldap.allUsernamesSearch.searchBase}" />
<constructor-arg index="1" value="${ldap.allUsernamesSearch.searchFilter}"
/>
</bean>
```

```
</constructor-arg>
<constructor-arg index="2">
<bean
class="org.pentaho.platform.plugin.services.security.userrole.ldap.transfor
m.SearchResultToAttrValueList">
<constructor-arg index="0"
value="${ldap.allUsernamesSearch.usernameAttribute}" />
</bean>
</constructor-arg>
</bean>
```

3. Replace it with this bean:

```
<bean id="allUsernamesSearch"
class="org.pentaho.platform.plugin.services.security.userrole.ldap.search.G
enericLdapSearch">
<constructor-arg index="0" ref="contextSource" />
<constructor-arg index="1">
<bean
class="org.pentaho.platform.plugin.services.security.userrole.ldap.search.L
dapSearchParamsFactoryImpl">
<constructor-arg index="0" value="${ldap.allUsernamesSearch.searchBase}" />
<constructor-arg index="1" value="${ldap.allUsernamesSearch.searchFilter}"
/>
<constructor-arg index="2">
<bean class="javax.naming.directory.SearchControls">
<!-- 2 comes from
http://java.sun.com/javase/6/docs/api/javax/naming/directory/SearchControls
.html#SUBTREE_SCOPE -->
<property name="searchScope" value="2" />
</bean>
</constructor-arg>
</bean>
</constructor-arg>
<constructor-arg index="2">
<bean
class="org.pentaho.platform.plugin.services.security.userrole.ldap.transfor
m.SearchResultToAttrValueList">
<constructor-arg index="0"
value="${ldap.allUsernamesSearch.usernameAttribute}" />
</bean>
</constructor-arg>
</bean>
```

4. Save the file and restart the Pentaho server.
5. Log into **PUC** and select any file or folder.

6. Click on **Folder Actions** > **Properties** > **Share**. From here, you can share folders by user or roles:

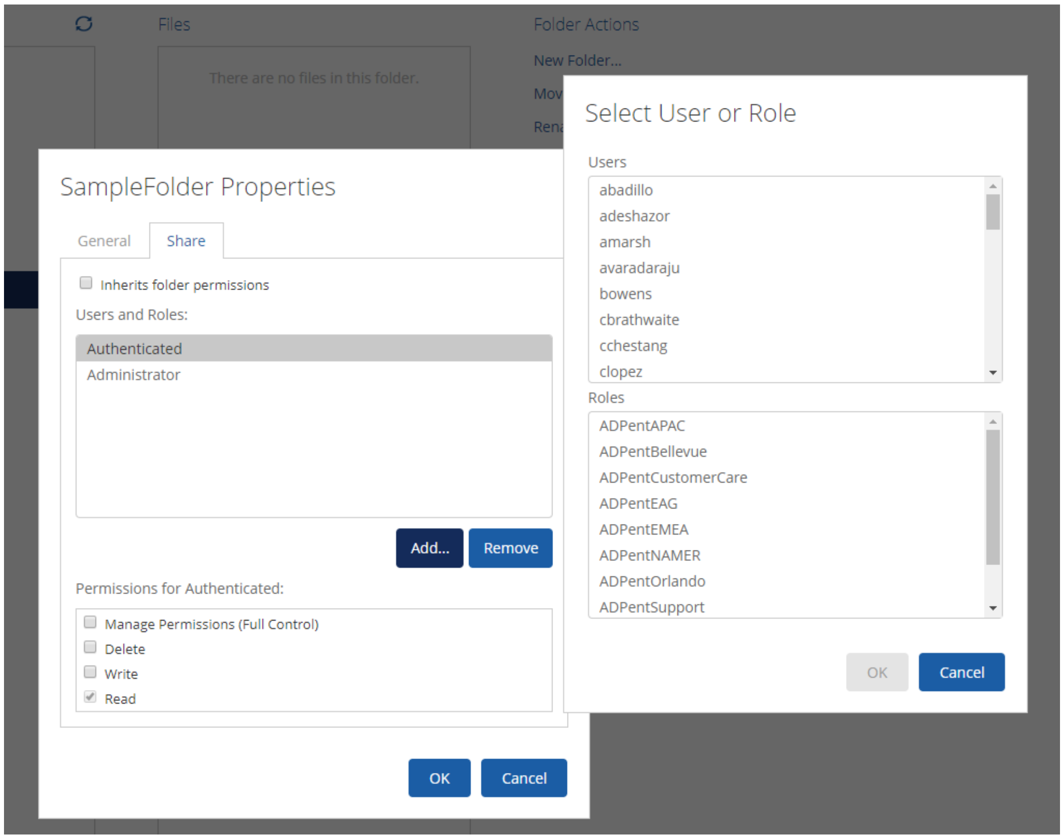


*Figure 11: Folder Properties*

## Configure Admin User for `pentaho.jms.cfg`

To configure the admin user for the `pentaho.jms.cfg`:

1. Locate the file `\pentaho-server\pentaho-solutions\system\karaf\etc\pentaho.jms.cfg`.
2. Change the following attributes to match those from your admin username in the Pentaho System Administrator section:

```
# The username and password of the broker receiving messages
userName=admin
password=password
```

## Related Information

Here are some links to information that you may find helpful while using this best practices document:

- Active Directory and Active Directory Domain Services Port Requirements
- Base64 encoder
- Enabling Verbose LDAP Logging
- LDAP Troubleshooting
- Manual LDAP Configuration
- Nested Roles or Groups (LDAP)
- Pentaho Components Reference
- Using Active Directory
- Spring Framework
- Switching to LDAP
- Switching to MS Active Directory

## Finalization Checklist

This checklist is designed to be added to any implemented project that uses this collection of best practices, to verify that all items have been considered and reviews have been performed.

Name of the Project:__________________________________________________

Date of the Review:___________________________________________________

Name of the Reviewer:________________________________________________

| Item | Response | Comments |
|---|---|---|
| Did you connect and bind to your LDAP/Active Directory server? | YES_______ NO_______ | |
| Did you configure the Pentaho system administrator? | YES_______ NO_______ | |
| Did you configure the other items listed in the Configuration section? | YES_______ NO_______ | |