# Contents

About This Document........................................................................................................ 4  
Purpose.................................................................................................................................. 4  
Scope..................................................................................................................................... 4  
Deployment Instructions ....................................................................................................... 5  
  Prerequisites ......................................................................................................................... 5  
  Copy the .jar Files.................................................................................................................. 5  
  Configure ECMA Script ......................................................................................................... 5  
  Configure Policies .................................................................................................................. 6  
  Configure Policies on Input Transformation of Publisher Channel ......................... 7  
  Configure Policies on Command Transformation of Subscriber Channel ...... 14  
  Configure Policies on Output Transformation of Subscriber Channel .......... 17  
Creating Custom Attributes for Storing Passlogix Call Status ....................... 18  
Creating Password Policies ................................................................................................. 19  
Configuring Property Files ................................................................................................. 20  
After Configuration............................................................................................................... 20  

Page 3 of 20
About This Document

Purpose
The purpose of this document is to provide detailed instructions for deploying the ESSO NIM integration solution.

Scope
The document covers deployment for the following use cases:

1. Create a new user in NIM, create an account in AD, then create a user and add user credentials in ESSO PG.
2. Modify a user’s password in NIM, then change the AD account password and propagate it to ESSO PG.
3. Disable a user in NIM, disable the AD account, and delete the user’s credentials from ESSO PG.
4. Enable a user in NIM, enable an AD account, and recreate the user’s credentials in ESSO PG.
5. Delete a user in NIM, delete an AD account, and delete the user and the user’s credentials from ESSO PG.
Deployment Instructions

Prerequisites

- The ESSO PG server and the ESSO PG Administrative Console must be installed.
- All components of NIM are installed, and you are able to do AD provisioning from the NIM Web console or iManager console.
- Creating users on NIM should be provisioned on AD; that is, eDirectory should be synchronized with AD by an AD Driver.
- The user with the same logon name as the user you are creating in NIM should be present in the ESSO PG repository, in case the repository for ESSO PG is not the root service.

Copy the .jar Files

Copy the following ESSO PG jar files to \NDS\lib\ on the NIM machine:

1. axis-1.2.1.jar
2. axis-ant-1.2.1.jar
3. bcprov-jdk13-128.jar
4. commons-discovery-0.2.jar
5. commons-logging-1.0.4.jar
6. jaxrpc.jar
7. log4j-1.2.9.jar
8. nimPasslogix.jar
9. opensaml-1.0.1.jar
10. PMCLI.jar
11. saaj.jar
12. wsdl4j-1.5.1.jar
13. wss4j.jar
14. xmlsec-1.3.0.jar

Remove j2evalidat.jar from \NDS\lib\ on the NIM machine.

Configure ECMA Script

ECMA Script functions will be called by policies which we configure on the publisher and subscriber channels of the respective target system driver. These ECMA Script functions will call appropriate the JAVA functions, which are deployed with NIMPasslogix.jar. Java functions will make appropriate calls to ESSO PG.

To configure the ECMA script:

1. Log on to iManager.
2. Navigate to Identity Manager > Identity Manager Overview > Driver Sets > AD Driver > Advanced > Insert.
3. Enter the name "Passlogix ECMA Script" and click OK.
4. Paste the script content in the dialog box and click Save.
**Configure Policies**

New policies are created as follows:

- Four new policies will be created and placed on input transformation of the publisher channel.
- Four new policies will be created and placed on command transformation of the subscriber channel.
- One new policy will be created and placed on output transformation of the subscriber channel.

These policies detect the presence of certain data in the XML data flowing between NIM and the target system and use that data as the criteria to make various calls to the ECMA script. These policies have
Deployment Instructions

to be configured on the subscriber and publisher channels for every target system driver, which is
supposed to be integrated with ESSO PG. Application names and login names need to be changed in
respective policies pertaining to different target systems.

Configure Policies on Input Transformation of Publisher Channel

1. Passlogix Add Credential Logic Policy

Passlogix add credential policies will be used to add credentials in ESSO PG when a new user is
created in NIM. After creation of an account on the target system, one association key will be
generated and then can be traced on the channel. This policy will check the presence of add-
association in XML data. If add-association is present, it queries the target account login name and
calls the ECMA script. The attribute to be queried is configurable, and can be specified in the policy
content. The ECMA script will make a call to ESSO PG. The content of this policy would be similar to
the example below.

1. Log on to iManager.
2. Navigate to Identity Manager > Identity Manager Overview > Driver Sets > AD Driver >
   Overview > Input Transformation Policies (of publisher channel) > Insert.
3. Enter the name 'Passlogix Add Credential Logic' and click OK.
4. Click Edit XML and paste the content (appearing below) in the dialog box.
5. Click Save.

"<?xml version="1.0" encoding="UTF-8"?>
<policy>
  <rule>
    <description>Passlogix Add Credential Logic</description>
    <conditions>
      <and>
        <if-operation op="equal">add-association</if-operation>
      </and>
    </conditions>
    <actions>
      <do-set-local-variable name="attrVal">
        <arg-string>
          <token-dest-attr name="CN"/>
        </arg-string>
      </do-set-local-variable>

      <!-- AD Server 2003 specified below is the application name. It can be different
      for different applications and would need to be changed on policies of the
      respective target system driver. -->

      <do-set-local-variable name="appName">
        <arg-string>
          <token-text xml:space="preserve">AD Server 2003</token-text>
        </arg-string>
      </do-set-local-variable>

      <do-set-local-variable name="userPwd" scope="driver">
        <arg-string>
          <token-dest-attr name="nspmDistributionPassword"/>
        </arg-string>
      </do-set-local-variable>
  </actions>
</rule>
</policy>
<do-set-local-variable name="completeAssociationUserId">
    <arg-string>
      <token-query datastore="src">
          <arg-association>
            <token-xpath expression="self::add-association"/>
          </arg-association>
        </token-query>
    </arg-string>
</do-set-local-variable>

<!-- the loginname specified below is the login attribute on a particular target system. The name of this attribute can vary from one target system to another. -->

<token-text xml:space="preserve">loginame</token-text>

<do-set-local-variable name="associationAttr">
    <arg-string>
      <token-xpath expression="self::add-association"/>
    </arg-string>
</do-set-local-variable>

<do-add-dest-attr-value name="PasslogixAddStatus">
    <arg-value type="string">
      <token-xpath expression="es:addPasslogixCred($attrVal,$appName,$userPwd,$completeAssociationUserId,$associationAttr)"/>
    </arg-value>
</do-add-dest-attr-value>

<do-set-local-variable name="isAddOperationExist" scope="driver">
    <arg-string>
      <token-text xml:space="preserve">N</token-text>
    </arg-string>
</do-set-local-variable>

</actions>
</rule>
""

2. Passlogix Modify Password Logic Policy

The Passlogix Modify Password policy is used to modify passwords in ESSO PG when a user's password is changed in NIM. This policy finds add-association keys and queries the target account login name. It then calls the ECMA script. The ECMA script will make a call to ESSO PG. The content of this policy would be similar to the example below.

1. Log on to iManager.
2. Navigate to Identity Manager > Identity Manager Overview > Driver Sets > AD Driver > Overview > Input Transformation Policies (of publisher channel) > Insert.
3. Enter the name 'Passlogix Modify Credential Logic' and click OK.
4. Click Edit XML and paste the content (appearing below) in the dialog box.
5. Click Save.
"<?xml version="1.0" encoding="UTF-8"?>
<policy>
    <rule>
        <description>Modify Password Logic</description>
        <conditions>
            <and>
                <if-operation mode="nocase" op="equal">status</if-operation>
                <if-xpath op="true">self::status[@level = 'success']/operation-data/password-subscribe-status/association[text() != "]</if-xpath>
            </and>
        </conditions>
        <actions>
            <do-set-local-variable name="attrVal">
                <arg-string>
                    <token-local-variable name="mdLocVarUID"/>
                </arg-string>
            </do-set-local-variable>
            <do-set-local-variable name="appName">
                <arg-string>
                    <token-text xml:space="preserve">AD Server 2003</token-text>
                </arg-string>
            </do-set-local-variable>
            <do-set-local-variable name="pwdVal">
                <arg-string>
                    <token-local-variable name="mdLocVarPassword"/>
                </arg-string>
            </do-set-local-variable>
            <do-set-local-variable name="completeAssociationUserId">
                <arg-string>
                    <token-query datastore="src">
                        <arg-association>
                            <token-local-variable name="modifyAssociation"/>
                        </arg-association>
                    </token-query>
                </arg-string>
            </do-set-local-variable>
        </actions>
        <!-- AD Server 2003 specified below is the application name. It can be different for different applications and would need to be changed on policies of the respective target system driver. -->
        <do-set-local-variable name="appVal">
            <arg-string>
                <token-text xml:space="preserve">AD Server 2003</token-text>
            </arg-string>
        </do-set-local-variable>
        <!-- the loginname specified below is the login attribute on a particular target system. The name of this attribute can vary from one target system to another. -->
        <arg-string>
            <token-text xml:space="preserve">loginname</token-text>
        </arg-string>
    </if-operation>
</policy>
3. Passlogix Disable User Account Policy

The Passlogix Disable User policy is used to delete credentials in ESSO PG when a user is disabled in NIM. It checks for add-association keys and queries the login attribute on the target system. It then calls the ECMA script. The ECMA script will make a call to ESSO PG. The content of this policy would be similar to the example below.

1. Log on to iManager.
2. Navigate to Identity Manager > Identity Manager Overview > Driver Sets > AD Driver > Overview > Input Transformation Policies (of publisher channel) > Insert.
3. Enter the name 'Passlogix Disable Credential Logic' and click OK.
4. Click Edit XML and paste the content (appearing below) in the dialog box.
5. Click Save.

```
<?xml version="1.0" encoding="UTF-8"?>
<policy>
  <rule>
    <description>Disable User Account Logic</description>
    <conditions>
      <and>
        <if-operation mode="nocase" op="equal">status</if-operation>
        <if-xpath op="true">self::status[@level = 'success']</if-xpath>
        <if-local-variable name="isDisablingAccount" op="equal">Y</if-local-variable>
      </and>
    </conditions>
  </rule>
</policy>
```
Deployment Instructions

<!-- AD Server 2003 specified below is the application name. It can be different for different applications and would need to be changed on policies of the respective target system driver. -->

<do-set-local-variable name="appName">
  <arg-string>
    <token-text xml:space="preserve">AD Server 2003</token-text>
  </arg-string>
</do-set-local-variable>

<do-set-local-variable name="pwdVal">
  <arg-string>
    <token-local-variable name="dsUserPwd"/>
  </arg-string>
</do-set-local-variable>

<do-set-local-variable name="completeAssociationUserId">
  <arg-string>
    <token-query datastore="src">
      <arg-association>
        <token-local-variable name="disableAssociation"/>
      </arg-association>
    </token-query>
  </arg-string>
</do-set-local-variable>

<do-add-dest-attr-value name="PasslogixDisableStatus">
  <arg-dn>
    <token-local-variable name="dsSrcDN"/>
  </arg-dn>
  <arg-value type="string">
    <token-xpath expression="es:deletePasslogixCred($attrVal,$appName,$pwdVal,$completeAssociationUserId,$associationAttr)"/>
  </arg-value>
</do-add-dest-attr-value>

<!-- the loginame specified below is the login attribute on a particular target system. The name of this attribute can vary from one target system to another. -->

<token-text xml:space="preserve">loginame</token-text>

<token-query>
</token-query>

<do-set-local-variable>
  <do-set-local-variable name="associationAttr">
    <arg-string>
      <token-local-variable name="disableAssociation"/>
    </arg-string>
  </do-set-local-variable>
</do-set-local-variable>

<do-add-dest-attr-value name="PasslogixDisableStatus">
  <arg-dn>
    <token-local-variable name="dsSrcDN"/>
  </arg-dn>
  <arg-value type="string">
    <token-xpath expression="es:deletePasslogixCred($attrVal,$appName,$pwdVal,$completeAssociationUserId,$associationAttr)"/>
  </arg-value>
</do-add-dest-attr-value>
4. Passlogix Enable User Account Policy

The Passlogix Enable User policy is used to recreate credentials in ESSO PG when a user is enabled in NIM. It checks for add-association keys and queries the login attribute on the target system. It then calls the ECMA script. The ECMA script will make a call to ESSO PG. The content of this policy would be similar to the example below.

1. Log on to iManager.
2. Navigate to Identity Manager > Identity Manager Overview > Driver Sets > AD Driver > Overview > Input Transformation Policies (of publisher channel) > Insert.
3. Enter the name 'Passlogix Enable Credential Logic' and click OK.
4. Click Edit XML and paste the content (appearing below) in the dialog box.
5. Click Save.

```
<?xml version="1.0" encoding="UTF-8"?>
<policy>
  <rule>
    <description>Enable Passlogix User Account Logic</description>
    <conditions>
      <and>
        <if-operation mode="nocase" op="equal">status</if-operation>
        <if-xpath op="true">self::status[@level = 'success']</if-xpath>
        <if-local-variable name="isEnablingAccount" op="equal">Y</if-local-variable>
      </and>
    </conditions>
    <actions>
      <do-set-local-variable name="attrVal">
        <arg-string>
          <token-local-variable name="enSrcDN"/>
        </arg-string>
      </do-set-local-variable>
      <!-- AD Server 2003 specified below is the application name. It can be different for different applications and would need to be changed on policies of the respective target system driver. -->
      <do-set-local-variable name="appName">
        <arg-string>
          <token-text xml:space="preserve">AD Server 2003</token-text>
        </arg-string>
      </do-set-local-variable>
    </actions>
  </rule>
</policy>
```
 Deployment Instructions

```xml
<do-set-local-variable name="userPwd">
  <arg-string>
    <token-local-variable name="enUserPwd"/>
  </arg-string>
</do-set-local-variable>

<do-set-local-variable name="completeAssociationUserId">
  <arg-string>
    <token-query datastore="src">
      <arg-association>
        <token-local-variable name="enableAssociation"/>
      </arg-association>
    </token-query>
  </arg-string>
</do-set-local-variable>

<do-set-local-variable name="associationAttr">
  <arg-string>
    <token-local-variable name="enableAssociation"/>
  </arg-string>
</do-set-local-variable>

<do-set-dest-attr-value name="PasslogixEnableStatus">
  <arg-dn>
    <token-local-variable name="enSrcDN"/>
  </arg-dn>
  <arg-value type="string">
    <token-xpath expression="es:enablePasslogixUser($attrVal,$appName,$userPwd,$completeAssociationUserId,$associationAttr)"/>
  </arg-value>
</do-set-dest-attr-value>

<do-set-local-variable name="isEnablingAccount" scope="driver">
  <arg-string>
    <token-text xml:space="preserve">N</token-text>
  </arg-string>
</do-set-local-variable>
```

<!-- the loginname specified below is the login attribute on a particular target system. The name of this attribute can vary from one target system to another. -->

```xml
	<token-text xml:space="preserve">loginname</token-text>
  </arg-string>
</token-query>
</arg-string>
</do-set-local-variable>

<do-set-local-variable name="associationAttr"/>
</do-set-local-variable>
<do-set-dest-attr-value name="PasslogixEnableStatus">
  <arg-dn>
    <token-local-variable name="enSrcDN"/>
  </arg-dn>
  <arg-value type="string">
    <token-xpath expression="es:enablePasslogixUser($attrVal,$appName,$userPwd,$completeAssociationUserId,$associationAttr)"/>
  </arg-value>
</do-set-dest-attr-value>

<do-set-local-variable name="isEnablingAccount" scope="driver">
  <arg-string>
    <token-text xml:space="preserve">N</token-text>
  </arg-string>
</do-set-local-variable>
```

Page 13 of 20
Configure Policies on Command Transformation of Subscriber Channel

1. Set Local Variable for Modify Password Policy

This policy will be created and placed on Command transformation of Subscriber channel. It sets the association key variable. The operation will be identified as Modify Password, and a variable called isModifyOperationExist is used by the policy on the publisher channel.

1. Log on to iManager.
2. Navigate to Identity Manager > Identity Manager Overview > Driver Sets > AD Driver > Overview > Command Transformation Policies (of Subscriber channel) > Insert.
3. Enter the name 'Set Local Variable for Modify Password Operation' and click OK.
4. Click Edit XML and paste the content (appearing below) in the dialog box.
5. Click Save.

"<?xml version="1.0" encoding="UTF-8"?>
<policy>
  <rule>
    <description>Set Local Variable for Password</description>
    <conditions>
      <or>
        <if-operation mode="regex" op="equal">modify-password</if-operation>
        <if-op-attr name="nspmDistributionPassword" op="changing"/>
      </or>
    </conditions>
    <actions>
      <do-set-local-variable name="mdLocVarUID" scope="driver">
        <arg-string>
          <token-xpath expression="@src-dn"/>
        </arg-string>
      </do-set-local-variable>
      <do-set-local-variable name="mdLocVarPassword" scope="driver">
        <arg-string>
          <token-attr name="nspmDistributionPassword"/>
        </arg-string>
      </do-set-local-variable>
      <do-set-local-variable name="isModifyOperationExist" scope="driver">
        <arg-string>
          <token-text xml:space="preserve">Y</token-text>
        </arg-string>
      </do-set-local-variable>
      <do-set-local-variable name="modifyAssociation" scope="driver">
        <arg-string>
          <token-association/>
        </arg-string>
      </do-set-local-variable>
    </actions>
  </rule>
</policy>"
2. Set Local Variable for User Delete Policy

This policy will be created and placed on Command transformation of the Subscriber channel for setting variables to be retrieved from the corresponding policy on the Publisher channel.

1. Log on to iManager.
2. Navigate to Identity Manager > Identity Manager Overview > Driver Sets > AD Driver > Overview > Command Transformation Policies (of Subscriber channel) > Insert.
3. Enter the name 'Set Local Variable for Delete Operation' and click OK.
4. Click Edit XML and paste the content (appearing below) in the dialog box.
5. Click Save.

```
"<?xml version="1.0" encoding="UTF-8"?>
<policy>
  <rule>
    <description>local variable setting for delete operation</description>
    <conditions>
      <and>
        <if-operation op="equal">delete</if-operation>
      </and>
    </conditions>
    <actions>
      <do-set-local-variable name="dlSrcDn" scope="driver">
        <arg-string>
          <token-xpath expression="@src-dn"/>
        </arg-string>
      </do-set-local-variable>
      <do-set-local-variable name="isDeleteUser" scope="driver">
        <arg-string>
          <token-text xml:space="preserve">Y</token-text>
        </arg-string>
      </do-set-local-variable>
    </actions>
  </rule>
</policy>
```

3. Set Local Variable for Enable Account Policy

This policy will be created and placed on Command transformation of the Subscriber channel for setting variables to be retrieved from corresponding policies on the Publisher channel. It verifies that the operation is 'modify'; that the source Login Disabled attribute is false; and that the destination Login Disabled is going to be changed. If all conditions are met, it sets the local variable `isEnablingAccount` to 'Y' to be used on the policy on the publisher channel. It also sets an association key.

1. Log on to iManager.
2. Navigate to Identity Manager > Identity Manager Overview > Driver Sets > AD Driver > Overview > Command Transformation Policies (of Subscriber channel) > Insert.
3. Enter the name 'Set Local Variable for Enable Operation' and click OK.
4. Click Edit XML and paste the content (appearing below) in the dialog box.
5. Click Save.

```
"<?xml version="1.0" encoding="UTF-8"?>
<policy>
  <rule>
    <description>Enable User</description>
    <conditions>
      <and>
        <if-operation op="equal">modify</if-operation>
      </and>
      <token-name expression="sourceLoginEnabled"/>
      <token-name expression="destinationLoginEnabled"/>
    </conditions>
    <actions>
      <do-set-local-variable name="isEnablingAccount" scope="driver">
        <arg-string>
          <token-text xml:space="preserve">Y</token-text>
        </arg-string>
      </do-set-local-variable>
    </actions>
  </rule>
</policy>
```
4. Set Local Variable for Disable Account Policy

This policy will be created and placed on Command transformation of the Subscriber channel for setting variables to be retrieved from corresponding policies on the Publisher channel. It verifies that the operation is 'modify'; that the source Login Disabled attribute is true; and that the destination Login Enabled is going to be changed. If all conditions are met, it sets the local variable isDisablingAccount to 'Y' to be used on the policy on the publisher channel. It also sets an association key.

1. Log on to iManager.
2. Navigate to Identity Manager > Identity Manager Overview > Driver Sets > AD Driver > Overview > Command Transformation Policies (of Subscriber channel) > Insert.
3. Enter the name 'Set Local Variable for Disable Operation' and click OK.
4. Click Edit XML and paste the content (appearing below) in the dialog box.
5. Click Save.

"<?xml version="1.0" encoding="UTF-8"?>
<policy>
  <rule>
    <description>set local var for disable account</description>
    <conditions>
      <and>
        <if-operation mode="nocase" op="equal">modify</if-operation>
        <if-src-attr name="Login Disabled" op="equal">false</if-src-attr>
        <if-op-attr name="Login Disabled" op="changing"/>
      </and>
    </conditions>
    <actions>
      <do-set-local-variable name="enUserPwd" scope="driver">
        <arg-string>
          <token-attr name="nspmDistributionPassword"/>
        </arg-string>
      </do-set-local-variable>
      <do-set-local-variable name="enSrcDN" scope="driver">
        <arg-string>
          <token-xpath expression="@src-dn"/>
        </arg-string>
      </do-set-local-variable>
      <do-set-local-variable name="isEnablingAccount" scope="driver">
        <arg-string>
          <token-text xml:space="preserve">Y</token-text>
        </arg-string>
      </do-set-local-variable>
      <do-set-local-variable name="enableAssociation" scope="driver">
        <arg-string>
          <token-association/>
        </arg-string>
      </do-set-local-variable>
    </actions>
  </rule>
</policy>"
Configure Policies on Output Transformation of Subscriber Channel

Passlogix Delete User Policy in Output Transformation Channel on Subscriber Channel

The Passlogix Delete User policy is used to delete a user and the user's credentials in ESSO PG when the user is deleted in NIM. The content of this policy would be similar to the example below.

1. Log on to iManager.
2. Navigate to Identity Manager > Identity Manager Overview > Driver Sets > AD Driver > Overview > Output Transformation Policies (of Subscriber channel) > Insert.
3. Enter the name Passlogix Delete Credential Logic' and click OK.
4. Click Edit XML and paste the content (appearing below) in the dialog box.
5. Click Save.

"<?xml version="1.0" encoding="UTF-8"?>
<policy>
  <rule>
    <description>Delete Passlogix User Logic</description>
    <conditions>
      <or>
        <if-operation op="equal">remove-association</if-operation>
        <if-operation mode="nocase" op="equal">delete</if-operation>
      </or>
    </conditions>
  </rule>
</policy>"
Creating Custom Attributes for Storing Passlogix Call Status

The following fields need to be added in NIM to show the return status from various ESSO PG calls. These fields are displayed individually for all ESSO PG applications for which credentials are added from NIM. The fields are:

- **PasslogixAddStatus.** Used to display the status when a user is created and credentials are added for an application.
- **PasslogixDisableStatus.** Used to display the status when a user is disabled and credentials are deleted for an application.
- **PasslogixEnableStatus.** Used to display the status when a user is enabled and credentials are recreated for an application.
- **PassLogixModPasswdStatus.** Used to display the status when a user’s password is modified and the password is changed for an application.

The fields will be created as below in NIM:

1. Log on to iManager.
3. For the Attribute name, enter PasslogixStatus.
4. Leave ASN1 ID blank. Click **Next**.
5. For Syntax, select 'Case Ignore String.'
6. Set attribute flags: select the check boxes for 'Public read' and 'Synchronized Immediately.'
7. Click **Finish**.
9. Select the attribute "PasslogixStatus" and move it to Optional attribute(s).
10. Navigate to Identity Manager > Identity Manager Overview > Driver Sets > ADDriver.
11. Click the green button.
12. Navigate to > Edit Property > select filter > select User Class.
13. Click 'Add Attribute.'
14. In the popup window, click **show all attributes** (if the created attribute is not displayed).
15. Select the check box for 'PasslogixStatus' and click **OK**.
16. Under User, click 'PasslogixStatus.'
17. On the right, select the 'synchronize' radio button for both the publisher and subscriber channels.

Repeat this process to create the PasslogixDisableStatus, PasslogixEnableStatus, and PassLogixModPasswdStatus attributes.

**Creating Password Policies**

You must create a password policy in NIM. To create the policy (if a simple password policy does not already exist):

1. Log on to iManager.
2. Click **Identity Manager**.
3. Click **Passwords**.
4. Click New. 'Step 1 of 8: Name and describe the Password Policy' appears.
5. Provide a name for the policy, for example, 'Passlogix password policy.'
6. Click **Next**.
7. At 'Would you like to enable Universal Password,' click **Yes**.
8. Uncheck the 'Enable the Advanced Password Rules' box.
9. At 'Step 2 of 8: Select the Universal Password options,' click **Next**.
10. At 'Step 3 of 8: Add rules to the Password Policy,' click **Next**.
11. At 'Step 4 of 8: Enable the Forgotten Password feature,' click **No**.
12. Click **Next**.
13. In the Assign To box, select SENA.
14. At 'Step 7 of 8: Assign the Password Policy,' click **Next**.
15. At 'Step 8 of 8: Summary of the Password Policy,' click **Finish**.
16. Assign this password policy to root context of eDirectory.

**Configuring Password Policies**

To configure password policies:

1. Click the created Password (or existing simple password policy).
2. In the pop-up windows, select the Universal Password tab.
3. Under this tab select 'Configuration Option.'
4. Verify that the following check boxes are selected:
   - Enable Universal Password.
   - Remove the NDS password when setting a Universal Password.
   - Synchronize Distribution Password when setting a Universal Password.
   - Allow a user to retrieve a password.
   - Allow an admin to retrieve a password.

   The other check boxes should be deselected.
5. Assign this password policy to root context of eDirectory.

Configuring Property Files

Create a folder in the C drive called "NIMProperties.". Paste "log4j.properties" and "PMClientConfiguration.properties" into this folder. In the PMClientConfiguration.properties file, change values for the following keys as required:

- javaCLI.serviceurl= <replace with the ESSO PG url>
- javaCLI.serviceuser= <replace with the ESSO PG admin id>
- javaCLI.serviceuserpassword= < replace with the ESSO PG admin password>

After Configuration

Restart the following components (in the order listed):

1. eDirectory.
2. tomcat.