RSA Identity Governance and Lifecycle Data Sheet for Novell eDirectory Connector

Version 1.1 | March 2017

Contents

Revision History

Revision Number	Description
Version 1.0	Novell eDirectory
Version 1.1	Added instructions for configuring the password vault with RSA Identity Governance and Lifecycle Novell eDirectory Connector

PURPOSE

This data sheet provides the configuration information required to create a new Novell eDirectory Connector.

SUPPORTED SOFTWARE

RSA Identity Governance and Lifecycle Version: *6.8.1 and above*

Application: *Novell eDirectory*

PREREQUISITES

Novell eDirectory should be installed and working properly.

SSL communication

For SSL communication, Novell eDirectory and certificate authority (CA) certificates (only if required) should be added to the appropriate keystores. Follow the steps mentioned below for adding certificates to the keystores of WebSphere, WebLogic and WildFly application servers:

WebSphere Application Server:

1. Log in to WebSphere Administrative console (http://<HOST_NAME>:9060/ibm/console/login.do)
2. In left panel, expand 'Security' menu.
3. Click 'SSL certificate' and 'key management' link.
4. Click 'Manage endpoint security configurations' link under 'Configuration Settings'.
5. Select 'Outbound' properties for the appropriate node.
6. Click appropriate node link to get the properties.
7. Under 'Related Items', click 'Key stores and certificates' and click the 'NodeDefaultTrustStore' keystore.
8. Under 'Additional Properties', click 'Signer certificates' and then click 'Retrieve from Port'.
9. In the 'Host' field, enter 'host name', enter 'SSL port(default-636)' in the 'Port' field, and 'eDir_cert' in the 'Alias' field.
10. Click 'Retrieve Signer Information'.
11. Verify that the certificate information is for the trusted certificate.
12. Click 'Apply' and 'Save'.
13. Login into WebSphere machine using SSH (example: putty).
14. On command prompt, run: /home/oracle/AFX/afx stop.
15. On command prompt, run: /opt/IBM/WebSphere/AppServer/bin/stopServer.sh server1.
16. On command prompt, run: /opt/IBM/WebSphere/AppServer/bin/startServer.sh server1.
17. On command prompt, run: /home/oracle/AFX/afx start.

WebLogic Application Server:

1. Download/Retrieve Novell eDirectory and CA (only if required) SSL certificates in PEM format and save them at location /home/oracle.
2. Log in to WebLogic Administrative console. (http://<HOST_NAME>.aveksa.local:7001/console/login/LoginForm.jsp)
3. Click 'Servers' link in 'Environment' section under 'Domain Configurations'.
4. Click 'aveksaServer' link.
5. Go to 'SSL' tab.
6. Click on 'Advanced' link.
7. Select 'HostName Verification = None'.
8. Save the settings.
9. Login into WebLogic machine using SSH (example: putty).
10. Change directory: cd /home/oracle/
11. Add the saved certificate and CA (only if required) certificates in server.keystore by using keytool
    Run: keytool -import -file <cert.pem> -alias <cert_alias> -keystore server.keystore
12. Restart SSL on WebLogic Server as mentioned below.
13. Go to Servers -> controls tab.
14. Select/check aveksaServer(admin) and then click Restart SSL.
15. Restart the Server. Go to /home/oracle/AFX/afx stop.
    Run:/home/oracle/wls/12.1.3.0/user_projects/domains/aveksaDomain/bin/stopWebLogic.sh
    Run:/home/oracle/wls/12.1.3.0/user_projects/domains/aveksaDomain/bin/startWebLogic.sh
    /home/oracle/AFX/afx start

WildFly Application Server:

1. Download/Retrieve Novell eDirectory and CA (only if required) SSL certificates in PEM format and save them.
2. cd <$JAVA_HOME>/jre/lib/security.
3. Add certificates in cacerts by using keytool
   keytool -import -file <cert.pem> -alias <cert_alias> -keystore cacerts

   Password for keystore (unless you have made any changes): changeit

   Run: keytool -import -file <certificate_path> -alias <certificate_alias> -keystore cacerts
4. Restart Server:
   Run:
   afx stop
   acm stop
   acm start
   afx start

MANAGE ENDPOINT CREDENTIALS USING A PASSWORD VAULT

To use a third-party password vault to manage the endpoint credentials, perform the following steps:

1. Configure the password vault according to the third-party provider's instructions.
2. Create a new password vault profile in the RSA Identity Governance and Lifecycle system for retrieving the Novell eDirectory password from the vault. See the RSA Identity Governance and Lifecycle Help for more information about creating a password vault profile.
3. Ensure that a Novell eDirectory account has been created at the configured password vault for storing the password.

Note: To use the dynamic password feature, step 1 must be completed. If a third-party password vault is not configured, configure the connector with a static password.

CONFIGURATION

The configuration of the Connector is completed through a number of screens. This section helps you to fill in the values for each screen.

General

The following table lists the parameters on the "General" screen, while creating the Connector.

Field Name	Value
Name	Novell eDirectory Connector
Description	Novell eDirectory LDAP Connector
Server	AFX Server
Connector Template	Novell eDirectory
State	Active
Export As Template	N/A

Settings – Connection Details

The following table lists the parameters on the "Settings-connection Details" screen, while creating the Connector.

Field Name	Value
Host	Host name or an IP address to the server
Port	Port Number to the server (Default 389)
Use Secure Connection	Wants secure connection or not (Default is false)
Login Distinguished Name	Distinguished Name to Login to the server
Static Password	Select this option, if you want to provide the password statically/manually. Fill in the value of password for the Novell eDirectory administrator user in the area provided.
Dynamic Password	Select this option to use a configured password vault to manage the endpoint credentials. After you select this option, either select a previously configured password vault profile from the drop-down menu, or click Create Profile to add a new password vault profile to use with this connector. RSA Identity Governance and Lifecycle uses this profile to retrieve the appropriate credentials from the password vault during connector deployment and connection tests.

Settings - Distinguished Name

The following table lists the parameters on the "Settings-Distinguished Name" screen, while creating the Connector.

Field Name	Value
Account DN Prefix	Prefix for Account Distinguished Name (Default is CN)
Account DN Suffix	Suffix for Account Distinguished Name
Group DN Prefix	Prefix for Group Distinguished Name
Group DN Suffix	Suffix for Group Distinguished Name

Settings - Object Creation

The following table lists the parameters on the "Settings-Object Creation" screen, while creating the Connector.

Field Name	Value
Object classes to create account	List of object classes to create Account (UserObjectClasses) e.g 'top','ndsLoginProperties','person','organizationalPerson','user'
Object classes to create group	List of object classes to create group (GroupObjectClasses) e.g 'top','group'

Settings – Group

The following table lists the parameters on the "Settings-Group" screen, while creating the Connector.

Field Name	Value
User membership attribute(s) for Group	List of membership attributes, give single quoted values seperated by comma - e.g. 'uniqueMember','equivalantToMe', default is 'member'
Use full DN for User membership attribute value	Wants to use full DN for User membership attribute value on Group object or not (Default is true)
Group membership attribute for User	List of membership attributes, give single quoted values seperated by comma – e.g. 'groupMembership','securityEquals' (Default is 'groupMembership')
Use full DN for Group membership attribute value	Wants to use full DN for Group membership attribute value on User object or not (Default is true)

Settings – AccountLockoutThreshold

The following table lists the parameters on the "Settings-Account Lockout Threshold" screen, while creating the Connector.

Field Name	Value
Intruder Login Attempts attribute value	Security setting determines the number of failed logon attempts that causes a user account to be locked out
Intruder Lockout Reset Interval attribute value(in seconds)	Up to this much time the account will be locked out. eg. Default :1800
Intruder Attempt Reset Interval attribute value(in seconds)	Designates the time frame in which to monitor consecutive failed login attempts. eg. Default :1800

Capabilities

The following commands are supported by the RSA Identity Governance and Lifecycle Novell eDirectory connector:

Create an Account

The following table lists the parameters on the "CreateAccount" screen.

Field Name	Value
Parameter Name	Account
Type	String
Default Value	-
Is the parameter required>	yes
Is the parameter encrypted>	no
Display Name	Account Name

Mapping	${User.User_Id}
Description	A full DN

Field Name	**Value**
Parameter Name	CN
Type	String
Default Value	-
Is the parameter required>	yes
Is the parameter encrypted>	no
Display Name	Common Name
Mapping	${User.User_Id}
Description	A name that represents an object. It is used to perform searches

Field Name	**Value**
Parameter Name	sn
Type	String
Default Value	-
Is the parameter required>	yes
Is the parameter encrypted>	no

Display Name	Last Name
Mapping	${User.Last_Name}
Description	Surname of a person

Field Name	**Value**
Parameter Name	givenName
Type	String
Default Value	-
Is the parameter required>	yes
Is the parameter encrypted>	no
Display Name	First Name
Mapping	${User.First_Name}
Description	Given name of a person

Field Name	**Value**
Parameter Name	mail
Type	String
Default Value	-
Is the parameter required>	yes

Is the parameter encrypted>	no
Display Name	Email address
Mapping	${User.Email_Address}
Description	Simple SMTP address of a person

Field Name	**Value**
Parameter Name	Password
Type	String
Default Value	-
Is the parameter required>	yes
Is the parameter encrypted>	no
Display Name	Initial password to set
Mapping	${AccountTemplate.Password}
Description	Password which is required for login

Delete Account

The following table lists the parameters on the "Delete Account" screen.

Field Name	**Value**
Parameter Name	Account
Type	String

Default Value	-
Is the parameter required>	yes
Is the parameter encrypted>	no
Display Name	Account Name
Mapping	${User.User_Id}
Description	A full DN

Reset Password

The following table lists the parameters on the "Reset Password" screen.

Field Name	**Value**
Parameter Name	Account
Type	String
Default Value	-
Is the parameter required>	yes
Is the parameter encrypted>	no
Display Name	Account Name
Mapping	${User.User_Id}
Description	A full DN

Field Name	Value
Parameter Name	Password
Type	String
Default Value	-
Is the parameter required>	yes
Is the parameter encrypted>	no
Display Name	Initial password to reset to
Mapping	${AccountTemplate.Password}
Description	A new password to reset

Enable Account

The following table lists the parameters on the "Enable an Account" screen.

Field Name	Value
Parameter Name	Account
Type	String
Default Value	-
Is the parameter required>	yes
Is the parameter encrypted>	no
Display Name	Account Name
Mapping	${User.User_Id}

Description	A full DN

Disable Account

The following table lists the parameters on the "Disable an Account" screen.

Field Name	**Value**
Parameter Name	Account
Type	String
Default Value	-
Is the parameter required>	yes
Is the parameter encrypted>	no
Display Name	Account Name
Mapping	${User.User_Id}
Description	A full DN

Lock Account

The following table lists the parameters on the "Lock an Account" screen.

Field Name	**Value**
Parameter Name	Account
Type	String
Default Value	-

Is the parameter required>	yes
Is the parameter encrypted>	no
Display Name	Account Name
Mapping	${User.User_Id}
Description	A full DN

Unlock Account

The following table lists the parameters on the "Unlock an Account" screen.

Field Name	**Value**
Parameter Name	Account
Type	String
Default Value	-
Is the parameter required>	yes
Is the parameter encrypted>	no
Display Name	Account Name
Mapping	${User.User_Id}
Description	A full DN

Move Account

The following table lists the parameters on the "Move Account" screen.

Field Name	**Value**

Parameter Name	Account
Type	String
Default Value	-
Is the parameter required>	yes
Is the parameter encrypted>	no
Display Name	Account Name
Mapping	${User.User_Id}
Description	A full DN

Field Name	**Value**
Parameter Name	NewParentDN
Type	String
Default Value	-
Is the parameter required>	yes
Is the parameter encrypted>	no
Display Name	New Parent's DN
Mapping	${AccountTemplate.NewParentDN}
Description	DN of new account base or organizational unit

Add Account To Group

The following table lists the parameters on the “Add an Account to Group” screen.

Field Name	Value
Parameter Name	Account
Type	String
Default Value	-
Is the parameter required>	yes
Is the parameter encrypted>	no
Display Name	Account Name
Mapping	${User.User_Id}
Description	A full DN

Field Name	Value
Parameter Name	Group
Type	String
Default Value	-
Is the parameter required>	yes
Is the parameter encrypted>	no
Display Name	Group DN or Group Name
Mapping	${Group.Name}

Description	A full DN of group or group name

Remove Account from Group

The following table lists the parameters on the "Remove Account from Group" screen.

Field Name	**Value**
Parameter Name	Account
Type	String
Default Value	-
Is the parameter required>	yes
Is the parameter encrypted>	no
Display Name	Account Name
Mapping	${User.User_Id}
Description	A full DN

Field Name	**Value**
Parameter Name	Group
Type	String
Default Value	-
Is the parameter required>	yes
Is the parameter encrypted>	no

Display Name	Group DN or Group Name
Mapping	${Group.Name}
Description	A full DN of group or group name

Update Account

The following table lists the parameters on the "Update an Account" screen.

Field Name	**Value**
Parameter Name	Account
Type	String
Default Value	-
Is the parameter required>	yes
Is the parameter encrypted>	no
Display Name	Account Name
Mapping	${User.User_Id}
Description	A full DN

Create Group

The following table lists the parameters on the "Create Group" screen.

Field Name	**Value**
Parameter Name	Group
Type	String

Default Value	-
Is the parameter required>	yes
Is the parameter encrypted>	no
Display Name	Group Name
Mapping	${Group.Name}
Description	A full DN

Field Name	**Value**
Parameter Name	CN
Type	String
Default Value	-
Is the parameter required>	yes
Is the parameter encrypted>	no
Display Name	Common Name
Mapping	${Group.CN}
Description	A name that represents an object. It is used to perform searches

Delete Group

The following table lists the parameters on the "Delete Group" screen.

Field Name	**Value**

Parameter Name	Group
Type	String
Default Value	-
Is the parameter required>	yes
Is the parameter encrypted>	no
Display Name	Group Name
Mapping	${Group.Name}
Description	A full DN

Update Group

The following table lists the parameters on the "Update Group" screen.

Field Name	**Value**
Parameter Name	Group
Type	String
Default Value	-
Is the parameter required>	yes
Is the parameter encrypted>	no
Display Name	Group Name
Mapping	${Group.Name}
Description	A full DN

APPENDIX

AFX Connector Binding

1. Go to Resource > Applications > <eDirectory Application Name> > AFX Connector Binding
2. Click "Edit Connector Binding"
3. AFX Connector: <eDirectory AFX Connector Name>

Enable AFX Fulfillment for Application

1. Go to Resource > Applications > <eDirectory Application Name> > Requests
2. Go to "Fulfillment" section
3. Set the Workflow to Default AFX Fulfillment

Limitations of Novell eDirectory AFX Adapter

- ***CreateAccount***: Account cannot be created when the AccountName (**CN** or **UID**) exceeds 32 characters.
- ***CreateAccount***: When account is created, that account can login to the server but can't view, modify or add data to the server.
- ***EnableAccount***: Enabling an already enabled account gives success.
- ***DisableAccount***: Disabling an already disabled account gives success.
- ***LockAccount***: Trying to lock an already-locked account gives success.
- ***UnlockAccount***: Trying to unlock an already-unlocked account gives success.
- **All Commands**: In the Account name and the Group name (Not full DN) characters mentioned in the http://www.ietf.org/rfc/rfc2253.txt are allowed, except for **"' (Single Quote)** and **'='**. Escaping these characters would not work if the account name or group name is not the full DN. You can provide these characters directly. If you want to use these characters in the Full DN of an account or Group, you have to use '\' before the character.

Configuration required for LockAccount and UnlockAccount commands

Accounts can get locked when any intruder tries to login after a certain number of failed attempts. Both the parent entry and the account require specific attributes for this security to take effect. To configure:

1. Log in to Novell eDirectory as the AdminUser.

2. Go to **Rights.**
3. Go to **View Effective Rights.**
4. Add the following attributes as effective rights to the parent entry: ***intruderAttemptResetInterval, loginIntruderLimit, loginIntruderAttempts, lockedByIntruder, loginIntruderResetTime.*** Accounts listed as trustees in the parent entry will inherit these attributes.
5. Set the following parent attributes' values**:**
    - ***intruderLockoutResetInterval*** **-** The time in seconds that the account remains locked.
    - ***lockoutAfterDetection***: Lock the account after intruder detection if set as "True."
    - ***detectIntruder*****:** Intruder detection will happen if set as "True."
    - ***intruderAttemptResetInterval***: Designates the time in seconds in which to monitor consecutive failed login attempts.
    - ***loginIntruderLimit*****:** The account will be locked when ***loginIntruderLimit*** of parent and /***loginIntruderAttempts*** of trustee account has the same value.

When an account is locked, the following attributes of the account are set:

- ***lockedByIntruder*****:** The account is locked due to intruder detection when set as "True".
- ***loginIntruderResetTime:*** Specifies the next time in seconds that the intruder attempts variable will be reset.
- ***loginIntruderAttempts*****:** Specifies the number of failed login attempts that have occurred in the current interval. This should be greater than or equal to the ***loginIntruderLimit*** of the parent entry when the account is locked.

References:

1. http://www.novell.com/communities/node/13705/locking-active-directory-account-when-locked-intruder-set-edirectory
2. http://ldapwiki.willeke.com/wiki/Locked%20By%20Intruder
3. http://www.novell.com/documentation/ndsv8/usnds/c1help/novell_ndsadmin/containerintruder.html

TROUBLESHOOTING

This section describes common errors when configuring the RSA Identity Governance and Lifecycle components for IBM ITDS and solutions.

- Dynamic password is used on connector settings page and the connector is not getting deployed with error displayed as: Password Vault Error.

  This may happen if there is an error in retrieving the password using the mentioned profile. You need to check if the values provided in the profile for password retrieval are valid and adequate permissions are present on the vault for fetching the password.

Copyrights

Trademarks