

**RSA Identity Governance and Lifecycle Connector Data Sheet
for
SQL Server**



Contents

PURPOSE	3
SUPPORTED SOFTWARE.....	3
PREREQUISITES.....	3
<i>MANAGE ENDPOINT CREDENTIALS USING A PASSWORD VAULT</i>	4
CONFIGURATION	4
<i>General</i>	4
<i>Settings</i>	5
<i>Capabilities</i>	5
<i>Command Input Parameters</i>	6
CreateAccount	6
DeleteAccount.....	7
ResetPassword.....	8
EnableAccount.....	9
DisableAccount.....	10
UpdateAccount.....	11
AddRoleToAccount.....	12
RemoveRoleFromAccount.....	13
TRUBLESHOOTING.....	15

PURPOSE

This data sheet provides the configuration information required to create a new SQL Server connector.

SUPPORTED SOFTWARE

- *RSA Identity Management and Governance 6.8.1 and later*
- *RSA Via Lifecycle and Governance 7.0.0 and later*
- *RSA Identity Governance and Lifecycle 7.0.1 and later*

Application: SQL Server

PREREQUISITES

Install SQL Server 2005

1. Install the SQL Server database against which you want to configure a collector
2. Download/get the driver from the respective vendor
3. Make sure that the downloaded jar should be present at or copied to following respective locations according to app server on the Aveksa Server or Remote Agent
 - For JBOSS :- <JBOSS_HOME>/standalone/deployments/aveksa.ear/APP-INF/lib
 - For Wildfly :- /tmp/repackaged/aveksa.ear/APP-INF/lib
 - For Websphere :- /opt/IBM/websphere/Apps/profiles/aveksaProfiles/installedApps/<host-node>/aveksa.ear/APP-INF/lib
 - For Weblogic :- /home/oracle/ACM-Weblogic
4. Ensure that the drive which contains the .jar file has the driver class file in it as well.
5. Now restart ACM. (Make sure that you do not have any requests in the queue)
6. Define the collector to use the database driver. If the driver is not in the available entries of the Database Type: then choose OTHER. Define the Driver Class and provide the URL of the given new driver

MANAGE ENDPOINT CREDENTIALS USING A PASSWORD VAULT

To use a third-party password vault to manage the endpoint credentials, perform the following steps. RSA Identity Governance and Lifecycle currently supports the password vault.

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 SQL server password from the vault. See the RSA Identity Governance and Lifecycle Help for more information about creating a password vault profile.
3. Ensure that an SQL server account has been created at the configured password vault for storing the 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	SQL Server Connector
Description	SQL Server Database Connector
Server	AFX Server
Connector Template	SQLServer2005
State	Active
Export As Template	N/A

Settings

The following table lists the parameters on the “Settings” screen, while creating the Connector.

Field Name	Value
Driver Class	com.microsoft.sqlserver.jdbc.SQLServerDriver
Driver URL	jdbc:sqlserver://[server]:[port]
User Name	Username for the SQL server administrator
Static Password	Select this option, if you want to provide the password manually and enter the password for the SQL server administrator.
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.

Capabilities

Commands:

- CreateAccount
- DeleteAccount
- ResetPassword
- EnableAccount
- DisableAccount
- UpdateAccount
- AddRoleToAccount
- RemoveRoleFromAccount

Command Input Parameters

CreateAccount

The following table lists the parameters on the “CreateAccount” screen.

Field Name	Value
Parameter Name	Account
Type	String
Default Value	None
Is the parameter required?	Yes
Is the parameter encrypted?	No
Display Name	Account Name
Mapping	\${AccountTemplate.Name}
Description	Parameter named Account expects to have Name of the account.

Field Name	Value
Parameter Name	Password
Type	Password
Default Value	None
Is the parameter required?	Yes

Is the parameter encrypted?	Yes
Display Name	Password
Mapping	`\${AccountTemplate.Password}`
Description	Parameter named Password expects to have Password of the account.

Field Name	Value
SQL Command	CREATE LOGIN `\${Account}` WITH PASSWORD = `\${Password}`

DeleteAccount

The following table lists the parameters on the “Delete Account” screen.

Field Name	Value
Parameter Name	Account
Type	String
Default Value	None
Is the parameter required?	Yes
Is the parameter encrypted?	No
Display Name	Account Name
Mapping	`\${Account.Name}`
Description	Parameter named Account expects to have Name of the account.

Field Name	Value
SQL Command	DROP LOGIN \${Account}

ResetPassword

The following table lists the parameters on the “Reset Password” screen.

Field Name	Value
Parameter Name	Account
Type	String
Default Value	None
Is the parameter required?	Yes
Is the parameter encrypted?	No
Display Name	Account Name
Mapping	\${Account.Name}
Description	Parameter named Account expects to have Name of the account.

Field Name	Value
Parameter Name	Password
Type	Password
Default Value	None

Is the parameter required?	Yes
Is the parameter encrypted?	Yes
Display Name	Password
Mapping	\${AccountTemplate.Password}
Description	Parameter named Password expects to have Password of the account.

Field Name	Value
SQL Command	ALTER LOGIN \${Account} WITH PASSWORD = '\${Password}'

EnableAccount

The following table lists the parameters on the “Enable Account” screen.

Field Name	Value
Parameter Name	Account
Type	String
Default Value	None
Is the parameter required?	Yes
Is the parameter encrypted?	No
Display Name	Account Name
Mapping	\${Account.Name}

Description	Parameter named Account expects to have Name of the account.
-------------	--

Field Name	Value
SQL Command	ALTER LOGIN \${Account} ENABLE

DisableAccount

The following table lists the parameters on the “Disable Account” screen.

Field Name	Value
Parameter Name	Account
Type	String
Default Value	None
Is the parameter required?	Yes
Is the parameter encrypted?	No
Display Name	Account Name
Mapping	\${Account.Name}
Description	Parameter named Account expects to have Name of the account.

Field Name	Value
SQL Command	ALTER LOGIN \${Account} DISABLE

UpdateAccount

The following table lists the parameters on the “Update Account” screen.

Field Name	Value
Parameter Name	Account
Type	String
Default Value	None
Is the parameter required?	Yes
Is the parameter encrypted?	No
Display Name	Account Name
Mapping	\${Account.Name}
Description	Parameter named Account expects to have Name of the account.

Field Name	Value
Parameter Name	Password
Type	Password
Default Value	None
Is the parameter required?	Yes
Is the parameter encrypted?	Yes
Display Name	Password

Mapping	`\${AccountTemplate.Password}`
Description	Parameter named Password expects to have Password of the account.

Field Name	Value
SQL Command	ALTER LOGIN `\${Account}` WITH PASSWORD = `\${Password}` MUST_CHANGE, CHECK_EXPIRATION=ON, CHECK_POLICY=ON

AddRoleToAccount

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

Field Name	Value
Parameter Name	Account
Type	String
Default Value	None
Is the parameter required?	Yes
Is the parameter encrypted?	No
Display Name	Account Name
Mapping	`\${Account.Name}`
Description	Parameter named Account expects to have Name of the account.

Field Name	Value
------------	-------

Parameter Name	AppRole
Type	String
Default Value	None
Is the parameter required?	Yes
Is the parameter encrypted?	No
Display Name	Application Role
Mapping	\${ApplicationRole.Name}
Description	Parameter named AppRole expects to have Application Role of the account.

Field Name	Value
SQL Command	EXEC master..sp_addsrvrolemember @loginame = \${Account}, @rolename = \${AppRole}

RemoveRoleFromAccount

The following table lists the parameters on the “Remove Role to Account” screen.

Field Name	Value
Parameter Name	Account
Type	String
Default Value	None
Is the parameter required?	Yes

Is the parameter encrypted?	No
Display Name	Account Name
Mapping	\${Account.Name}
Description	Parameter named Account expects to have Name of the account.

Field Name	Value
Parameter Name	AppRole
Type	String
Default Value	None
Is the parameter required?	Yes
Is the parameter encrypted?	No
Display Name	Application Role
Mapping	\${ApplicationRole.Name}
Description	Parameter named AppRole expects to have Application Role of the account.

Field Name	Value
SQL Command	EXEC master..sp_dropsvrolemember @loginame = \${Account}, @rolename = \${AppRole}

TROUBLESHOOTING

This section describes common errors when configuring the RSA Identity Governance and Lifecycle components for SQL server 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

Copyright © 2017 EMC Corporation. All Rights Reserved. Published in the USA.

TRADEMARKS

RSA, the RSA Logo, and EMC are either registered trademarks or trademarks of EMC Corporation in the United States and/or other countries. All other trademarks used herein are the property of their respective owners. For a list of EMC trademarks, go to www.rsa.com/legal/trademarks_list.pdf.