RSA Identity Governance and Lifecycle Collector Data Sheet
for
IBM DB2

Version 1.1 | August 2017

Table of Contents

Revision History

Revision Number	**Description**
Version 1.0	DB2 Data Collector creation and configuration
Version 1.1	Added instructions for configuring the password vault with RSA Identity Governance and Lifecycle ,DB2 Database , Identity Data Collector, Account Data Collector and Entitlement Data Collector

Purpose

This data sheet provides the configuration information required to create a new Identity data collector, Account data collector and Entitlement data collector for DB2.

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: IBM DB2 10.5 and later

Collector Type(s): Identity Data Collector, Account Data Collector, Entitlement Data Collector

Prerequisites

1. Install the DB2 database against which you want to configure the collector
2. Download/get the driver from the respective vendor
    - For DB2 – download db2jcc.jar
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, WAS and WebLogic ,Refer steps mentioned below for respective servers
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

WildFly

Customize RSA Identity Governance and Lifecycle

On an RSA Identity Governance and Lifecycle appliance or software appliance running v7.0 or later, you must use the customizeACM.sh tool to extract the aveksa.ear file, customize it, and repackage it. The following procedure requires that you know how to use the customizeACM.sh tool.

You can customize RSA Identity Governance and Lifecycle by modifying the aveksa.ear file located at /home/oracle/archive.

RSA provides a utility (customizeACM.sh in /home/oracle/deploy) that allows you to extract the aveksa.ear file and rebuild a customized version. For more information, see "Customize RSA Identity Governance and Lifecycle" in the Installation Guide.

Procedure

1. Log on to the appliance as the admin user using an SSH tool, such as Putty.
2. Verify that RSA Identity Governance and Lifecycle is running. Enter the following command:
    - sudo service aveksa_server status
3. If the server is running, the following message appears:
    - RSA Identity Governance and Lifecycle Compliance Manager Server is running.
4. If the message indicates that the server is not running, enter the following command:
    - sudo service aveksa_server start
5. Change to the oracle user. Enter the command:
    - su – oracle
6. Go to /home/oracle/deploy.
7. Run the customizeACM.sh script to extract the .ear file to modify and specify its location. Enter the command: customizeACM.sh -c <*path to the ear file*>
    Note: If you do not specify the path to the .ear file, the script prompts you to use the currently deployed .ear file.
    If you want to use the currently deployed .ear, enter 'yes'
    Otherwise, enter 'no'
8. Content of this .ear file will be extracted to a directory at the following location: /tmp/customizeACM/
9. Go to /tmp/customizeACM/ and copy db2jcc.jar to /tmp/customizeACM/APP-INF/lib
10. After completing the file modification, run the customizeACM.sh script again to rebuild the .ear file. Go to /home/oracle/deploy, enter the command:
    - ./customizeACM.sh –d
11. The script deploys the new customized .ear file and archives it to the following location, appending time and date stamp to name: /home/oracle/archive

Note: You do not need to restart ACM or AFX.

WebSphere

1. Copy the db2jcc.jar to the "lib" directories Under APP-INF folder:
    - /opt/IBM/WebSphere/AppServer/profiles/aveksaProfile/installedApps/<*HostName*>Node01Cell/aveksa.ear /APP-INF/lib
2. Restart the WebSphere Application Server.

WebLogic

1. Copy the db2jcc.jar files to following location and restart the server.
    - Location for aveksa.ear: /home/oracle/ACM-WebLogic
2. Create a new folder: mkdir /tmp/aveksa.ear
3. Unzip aveksa.ear to /home/oracle/ACM-WebLogic
    - Example: unzip -q -X /home/oracle/ACM-WebLogic/aveksa.ear -d /tmp/aveksa.ear
4. Rename existing aveksa.ear file to aveksa.ear.old in /home/oracle/ACM-WebLogic/
    - Example: mv /home/oracle/ACM-WebLogic/aveksa.ear /home/oracle/ACM-WebLogic/aveksa.ear.old
5. Copy db2jcc.jar file to /tmp/aveksa.ear/APP-INF/lib/
6. Repackage aveksa.ear
    - Note that you are creating the ear file at a new location: /home/oracle/ACM-WebLogic
    - cd /tmp/aveksa.ear
    - zip -q -r -u /home/oracle/ACM-WebLogic/aveksa.ear *

7. Deploy new ear file.
   - Login to the WebLogic Admin console:
     Example: http://<*server_hostname*>:7001/console
   - Go to Deployment > Server.
   - Select aveksa.ear and click the Update button.
   - Select path /home/oracle/ACM-WebLogic and select aveksa.ear.
   - Click Next and then Finish.
   - Restart the WebLogic Server.

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 DB2 user 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 DB2 user account has been created at the configured password vault to store the password.

Identity Data Collector

Configuration

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

Collector Description

The following table lists the parameters on the "Collector Description" screen.

Field Name	**Value**
Collector Name	DB2 Identity Data Collector
Description	Provide a description for this Collector.
Data Source Type	Select "Database" from the list.
Agent	Select any available Agent. The default is AveksaAgent.
Directory	Select Database Directory from the drop-down menu. If Database Directory doesn't exist, create a new directory from the Resources > Directories menu.

Status	Set status as Active.
Copy from	If a configured Identity Collector exists, all of its parameters can be copied. Select it from the list to copy it.
Scheduled	This parameter schedules runs of this Collector instance. Set Start and Frequency when selecting this option.

Configuration Information

The following table lists the parameters on the "Configuration Information" screen.

Field Name	**Value**
DB Type	*Choose DB type as a database configured in prerequisites steps if it is present in the already defined list else select "Custom"*
Driver Class	You can get this information from the documentation of this driver i.e. jar file downloaded for this collector. For e.g. COM.ibm.db2os390.sqlj.jdbc.DB2SQLJDriver
URL	Syntax is like jdbc:[subprotocol]: [subsubprotocol:][databasename][;attributes]. For any database you need to provide the schema or database name and the port assigned to this one at the time of its creation. For e.g. jdbc:db2:@//<Hostname where database is running>:<Database port>/<Database name>
User Name	Username to login to database (Make sure that this user has all privileges on these tables for e.g. "sys as sysdba")
Static Password	Select this option to provide the password manually and enter the password for the DB2 user.
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, 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.

Select types of account data to collect

Select 'Users' as identity data type as applicable.

Mapping for user attributes

User Data: The following table lists the parameters on the "Mapping for user attributes" screen.

Field Name	**Value**
Users Data Query	(Required). Query to return user attribute values. The column names resulting from the query will be used in the fields. Example: select user_id, first_name, last_name, email, supervisor from t_users
User ID	(Required). User ID column name resulting from <Users Data Query>.
Admin	Admin column name resulting from <Users Data Query>. And select respective value is User ID, Name, Title etc. of User
Business Unit Id	Business Unit Id column name resulting from <Users Data Query>. And select Business Unit Id value is Name or Backup Business Owner or Backup Technical Owner of Business Unit from drop down
Backup Supervisor	Backup Supervisor column name resulting from <Users Data Query>. And select respective value is User ID, Name, Title etc. of User
Business Unit Admin	Business Unit Admin column name resulting from <Users Data Query>. And select respective value is User ID, Name, Title etc. of User
Department	Department column name resulting from <Users Data Query>.
Email Address	Email Address column name resulting from <Users Data Query>.
First Name	First Name column name resulting from <Users Data Query>.
Is Terminated	Is Terminated column name resulting from <Users Data Query>.
Job Code	Job code column name resulting from <Users Data Query>.
Job Status	Job Status column name resulting from <Users Data Query>.

Join Date	Join Date column name resulting from <Users Data Query>.
Last Name	Last Name column name resulting from <Users Data Query>.
Location	Location column name resulting from <Users Data Query>.
Supervisor	Supervisor column name resulting from <Users Data Query>.
Technical Advisor	Technical Advisor column name resulting from <Users Data Query>. And select respective value is User ID, Name, Title etc. of User.
Termination Date	Termination Date column name resulting from <Users Data Query>.
Title	Title column name resulting from <Users Data Query>.
Unique ID	Unique ID column name resulting from <Users Data Query>.

Account Data Collector

Configuration

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

Collector Description

The following table lists the parameters on the “Collector Description” screen.

Field Name	**Value**
Collector Name	DB2 Account Data Collector
Description	Description for ADC (For example: DB2 Account Collector)
Business Source	Select Database from available list
Data Source Type	Select the required application
Agent	Appropriate agent (By default, AveksaAgent)
Status	Active
Copy from	The existing Database collector from which to copy a configuration (by default, this is blank)
Scheduled	Select yes to schedule collection

Configuration Information

The following table lists the parameters on the “Configuration Information” screen.

Field Name	**Value**
DB Type	Choose DB type as a database configured in prerequisites steps if it is present in the already defined list, else select “Custom”
Driver Class	You can get this information from the documentation of this driver i.e. jar file downloaded for this collector. For e.g. COM.ibm.db2os390.sqlj.jdbc.DB2SQLJDriver

URL	Syntax is like jdbc:[subprotocol]: [subsubprotocol:][databasename][;attributes]. For any database you need to provide the schema or database name and the port assigned to this one at the time of its creation. For e.g. jdbc:db2:@//<Hostname where database is running>:<Database port>/<Database name>
User Name	Username to login to database (Make sure that this user has all privileges on these tables for e.g. “sys as sysdba”)
Static Password	Select this option to provide the password manually and enter the password for the DB2 user.
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, 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.

Select types of account data to collect

You can select multiple account data types such as Accounts, User Account Mappings, Groups and Sub Groups as applicable.

Mapping for account attributes

The following table lists the parameters on the “Mapping for account attributes” screen.

Account Data

Field Name	**Value**
Accounts Data Query	(Required). Query to return account data. The column names resulting from the query will be used in the fields. Example: select account,last_login_date from t_accounts
Account ID/Name	(Required). Account ID or Name column name resulting from <Accounts Data Query>.
Last Login Date	Last Login Date column name resulting from <Accounts Data Query>.

Expiration Date	Expiration Date column name resulting from <Accounts Data Query>.

Mapping for user account mapping attributes

The following table lists the parameters on the “Mapping for user account mapping attributes” screen.

User Account Mappings Data

Field Name	**Value**
User Account Mappings Data Query	(Required). Query to return user account mapping data. The column names resulting from the query will be used in the fields. Example: select account,user from t_user_account_mappings
User ID	(Required). User ID column name resulting from User <Account Mappings Data Query>.
Account ID/Name	Account ID or Name column name resulting from User <Account Mappings Data Query>.

Mapping for group attributes

The following table lists the parameters on the “Mapping for group attributes” screen.

Group Data

Field Name	**Value**
Groups Data Query	(Required). Query to return group attribute values. The column names resulting from the query will be used in the fields. Example: select group_id, description from t_groups
Group ID/ Name	(Required). Group ID or Name column name resulting from <Groups Data Query>.
Group admin	Group admin column name resulting from <Groups Data Query>.
Owner	(Required). Owner column name resulting from <Groups Data Query>.

Account Membership Data

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

Field Name	Value
Account Membership Query	(Required). Query to return account members of groups. The column names resulting from the query will be used in the fields. Example: select account_id, group_id from t_group_memberships where type = 'account'.
Account ID/Name	(Required). Account ID or Name column name resulting from <Account Membership Query>.
Group ID/ Name	(Required). Group ID or Name column name resulting from <Account Membership Query>.

Mapping for subgroup attributes

The following table lists the parameters on the “Mapping for subgroup attributes” screen.

Subgroup Data

Field Name	Value
Subgroup Membership Query	(Required). Query to return sub-group members of groups. The column names resulting from the query will be used in the fields. Example: select sub_grp_id, group_id from t_group_memberships where type = 'group'
Subgroup ID/Name	(Required). Subgroup ID or Name column name resulting from <Subgroup Membership Query>.
Group ID/ Name	(Required). Group ID or Name column name resulting from <Subgroup Membership Query>.

Entitlement Data Collector

Configuration

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

Collector Description

The following table lists the parameters on the “Collector Description” screen.

Field Name	**Value**
Collector Name	Name for the Database Entitlement Collector.
Description	Description for the EDC. Example, Database Entitlement Collector
Data Source Type	Select Database Directory.
Business Source	Select the required application.
Agent	Appropriate agent (by default, AveksaAgent)
Status	Active
Copy from	Select existing Database Entitlement collector if you want to copy its configuration. (This is blank by default.)
Scheduled	Select yes to schedule collection

Configuration Information

The following table lists the parameters on the “Configuration Information” screen.

Field Name	**Value**
DB Type	*Choose DB type as a database configured in prerequisites steps if it is present in the already defined list else select “Custom”*

Driver Class	You can get this information from the documentation of this driver i.e. jar file downloaded for this collector. For e.g. COM.ibm.db2os390.sqlj.jdbc.DB2SQLJDriver
URL	Syntax is like jdbc:[subprotocol]: [subsubprotocol:][databasename][;attributes]. For any database you need to provide the schema or database name and the port assigned to this one at the time of its creation. For e.g. jdbc:db2@//<Hostname where database is running>:<Database port>/<Database name>
User Name	Username to login to database (Make sure that this user has all privileges on these tables for e.g. “sys as sysdba”)
Static Password	Select this option to provide the password manually and enter the password for the DB2 user.
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, 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.

Select types of entitlement data to collect

Collect resource-action and application role entitlements for Groups, Accounts and Users. You can select multiple entitlements data type for this option as applicable.

Define General Column Names

The following table lists the parameters on the “Define General Column Names” screen.

Field Name	**Value**
User Reference ID/Name	Common User Reference ID or Name, column name returned by several queries
Resource Fully Qualified Name	Common Resource Fully Qualified Name
Action ID/Name	Common Action ID or Name, column name returned by several queries

Application Role ID/Name	Common Application Role ID or Name column name, returned by several queries

Mapping for resource attributes

The following table lists the parameters on the “Mapping for resource attributes” screen, while creating the Collector.

Resource Data

Field Name	**Value**
Resources Data Query	Query to return resource attribute values for resource-action entitlements. The column names resulting from the query will be used in the fields below. Example: select resource as FQN, Owner, Location from t_resources
Resource ID/Name	Resource ID or Name column name resulting from <Resources Data Query>.
Resource Fully Qualified Name	Resource Fully Qualified Name defined in Generic Column Names

Mapping for resource-action based entitlements

The following table lists the parameters on the “Mapping for resource-action based entitlements” screen.

Resource Entitlement Data

Field Name	**Value**
Resource Entitlements Query	Query to return entitlement attribute values for resource-action entitlements. Example: select distinct resource as FQN, action from t_resource_ents
Resource Fully Qualified Name	Resource Fully Qualified Name defined in Generic Column Names.
Action ID/Name	Action ID/Name defined in Generic Column Names.

User Data

The following table lists the parameters on the “User Data” screen, while creating the Collector.

Field Name	Value
Ents. for Users Query	Query to return resource-action entitlements granted to users. Example: select resource as FQN, action, user_id from t_resource_ents where type = 'user'
Entitled User	User Reference ID/Name defined in Generic Column Names
Resource Fully Qualified Name	Resource Fully Qualified Name defined in Generic Column Names.
Action ID/Name	Action ID/Name defined in Generic Column Names.

Group Data

The following table lists the parameters on the “Group Data” screen, while creating the Collector.

Field Name	Value
Ents. For Groups Query	Query to return resource-action entitlements granted to groups. Example: select resource as FQN, action, user_id from t_resource_ents where type = 'group'
Entitled Group	User Reference ID/Name defined in Generic Column Names
Resource Fully Qualified Name	Resource Fully Qualified Name defined in Generic Column Names.
Action ID/Name	Action ID/Name defined in Generic Column Names.

Account Data

The following table lists the parameters on the “Account Data” screen, while creating the Collector.

Field Name	Value
Ents. For Accounts Query	Query to return resource-action entitlements granted to user accounts. Example: select resource as FQN, action, user_id from t_resource_ents where type = 'account'

Entitled Account	User Reference ID/Name defined in Generic Column Names
Resource Fully Qualified Name	Resource Fully Qualified Name defined in Generic Column Names.
Action ID/Name	Action ID/Name defined in Generic Column Names.

Mapping for application role attributes

Application Role Data

The following table lists the parameters on the “Mapping for application role attributes” screen, while creating the Collector.

Field Name	**Value**
Application Roles Query	Query to return application role attribute values for application-role entitlements. Example: select distinct approle from t_approle_defs
Application Role ID/Name	Application Role ID/Name defined in Generic Column Names

Resource-Action Entitlements Data

The following table lists the parameters on the “Resource-Action Entitlement Data” screen, while creating the Collector.

Field Name	**Value**
Resource-Action Entitlements of App Roles Query	Query to return resource-action entitlement sub-components of application role entitlements that were collected above. Example: select approle_parent as approle, resource as FQN, action from t_approle_members where type = 'resource'
Application Role ID/Name	Application Role ID/Name defined in Generic Column Names
Resource Fully Qualified Name	Resource Fully Qualified Name defined in Generic Column Names.
Action ID/Name	Action ID/Name defined in Generic Column Names.

Child Application Roles Data

The following table lists the parameters on the “Child Application Roles Data” screen, while creating the Collector.

Field Name	**Value**
Child App Roles of App Roles Query	Query to return application role entitlement children of application role entitlements that were collected above. Example: select approle_parent as approle, approle_child from t_approle_members where type = 'app-role'
Child Application Role ID/Name	Child Application Role ID or Name column name resulting from Child App Roles of App Roles Query
Application Role ID/Name	Application Role ID/Name defined in Generic Column Names

Mapping for application role based entitlements

The following tables lists the parameters on the “Mapping for application role based entitlements” screen.

Group Data

The following table lists the parameters on the “Group Data” screen.

Field Name	**Value**
App Roles for Groups Query	Query to return application role entitlements granted to groups. Example: select approle, user_id from t_approle_ents where type = 'group'
Entitled Group	User Reference ID/Name defined in Generic Column Names
Application Role ID/Name	Application Role ID/Name defined in Generic Column Names

Account Data

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

Field Name	**Value**
App Roles for Accounts Query	Query to return application role entitlements granted to user accounts. Example: select approle, user_id from t_approle_ents where type = 'account'

Entitled Account	User Reference ID/Name defined in Generic Column Names
Application Role ID/Name	Application Role ID/Name defined in Generic Column Names

User Data

The following table lists the parameters on the “User Data” screen.

Field Name	**Value**
App Roles for Users Query	Query to return application role entitlements granted to users. Example: select approle, user_id from t_approle_ents where type = 'user'
Entitled User	User Reference ID/Name defined in Generic Column Names
Application Role ID/Name	Application Role ID/Name defined in Generic Column Names

COPYRIGHTS

TRADEMARKS