

**RSA Identity Governance and Lifecycle  
Connector Data Sheet  
for  
Red Hat Enterprise Linux (RHEL)**



## Table of Contents

<b>Purpose</b> .....	<b>4</b>
<b>Supported Software</b> .....	<b>4</b>
<b>Prerequisites</b> .....	<b>4</b>
<b>Configuration</b> .....	<b>5</b>
<i>General</i> .....	5
<i>Settings</i> .....	6
<i>Capabilities</i> .....	6
Command .....	6
1. Create an account .....	7
2. Update an account .....	8
3. Reset account's password .....	10
4. Add an account to a group .....	11
5. Remove an account from a group .....	12
6. Enable an account .....	13
7. Disable an account .....	14
8. Create a group .....	15
9. Delete a group .....	16

## Revision History

<b>Revision Number</b>	<b>Description</b>
Version 1.0	RHEL Linux and added instructions for configuring the RSA Identity Governance and Lifecycle RHEL Linux Connector.
Version 1.1	Updated document with minimum version support.

## Purpose

This data sheet provides the configuration information required to create a new RHEL *Linux* connector.

## Supported Software

- *RSA Identity Management and Governance 6.9.1 and later*
- *RSA Via Lifecycle and Governance 7.0.0 and later*
- *RSA Identity Governance and Lifecycle 7.0.1 and later.*

**Application:** *Red Hat Enterprise Linux 6.3*

## Prerequisites

SSH (sshd) server should be running on the host RHEL machine.

**Note:** By default this server is up and running on the Linux machine.

## Configuration

The Connector creation is made up of three sections:

**General** – General details about the Connector; such as the name, type etc.

**Settings** – The connection settings required to connect the RSA-IMG and the End-point Application in consideration.

**Capabilities** – These are the list of “verbs” or capabilities that the RSA-IMG Connector supports; for example Create, Update, Delete, etc.

### General

The following table helps you with the Parameters asked on the “General” screen while creating the Connector.

Field Name	Value
Name	RHEL Linux Connector
Description	RHEL Linux Connector
Server	AFX Server
Connector Template	<i>Linux</i>
State	Test
Export As Template	N/A

**Note:** When you are satisfied your connector is configured properly change the state to Active. No automated provisioning will occur while in the Test state. It is recommended that you test all enabled commands using the Test Connector Settings and Test Connector Capabilities prior to changing to the Active state.

### Settings

The following table helps you with the Parameters asked on the “Settings” screen while creating the Connector.

Field Name	Value
<i>Host</i>	<Hostname or an IP address of the Linux machine>(e.g. m/c name mymachine or localhost or IP like 192.30.30.124)
<i>Port</i>	<SSH Port Number to the Linux machine>(e.g. 22)
<i>Timeout (milliseconds)</i>	<Connection timeout(milliseconds)> (Default 5000)
<i>Login Name</i>	<i>Login name for the Linux machine</i>
<i>Password</i>	<i>Password to login to the Linux machine</i>

## Capabilities

The following table(s) helps you with the Parameters asked when enabling the commands (verbs) on the “Capabilities” screen; while creating the Connector.

Following commands are supported by Linux RHEL Connector:

### Command

- Create an account
- Update an account
- Reset account’s password
- Add an account to a group
- Remove an account from a group
- Enable an account
- Disable an account
- Create a group
- Delete a group

## Command Input Parameters

### 1. Create an account

While defining the “Create an Account” verb, certain parameters need to be defined. The following tables will help you define these parameters. Each table is dedicated to one parameter.

Field Name	Value
Parameter Name	<i>Account</i>
Type	<i>STRING</i>
Default Value	<i>None</i>
Is the parameter required?	Yes
Is the parameter encrypted?	No
Display Name	<i>Account Name</i>
Mapping	<i>\${Account.Name}</i>
Description	<i>&lt;Name of the account&gt;</i>

Field Name	Value
Parameter Name	<i>Password</i>
Type	<i>STRING</i>
Default Value	<i>None</i>
Is the parameter required?	Yes
Is the parameter encrypted?	Yes
Display Name	<i>Password</i>
Mapping	<i>\${AccountTemplate.Password}</i>
Description	<i>&lt;Password for account to be created&gt;</i>

## Command Code

Field Name	Value
Shell Command	useradd \${Account} -p `perl -e 'print crypt('\${Password}', 'salt')` 2>&1

**Note** : Password containing ` or \$ will work only if \ is appended before it. e.g To use Abc\$%^ as password, add it as Abc\\$%^.

## 2. Update an account

While defining the “Update an Account” verb, certain parameters need to be defined. The following tables will help you define these parameters. Each table is dedicated to one parameter.

### Command Input Parameters

Field Name	Value
Parameter Name	<i>Account</i>
Type	<i>STRING</i>
Default Value	<i>None</i>
Is the parameter required?	Yes
Is the parameter encrypted?	No
Display Name	<i>Account Name</i>
Mapping	<i>\${Account.Name}</i>
Description	<i>&lt;Name of the account&gt;</i>

Field Name	Value
Parameter Name	<i>Home</i>



Type	<i>STRING</i>
Default Value	<i>None</i>
Is the parameter required?	Yes
Is the parameter encrypted?	No
Display Name	<i>Home Directory</i>
Mapping	<i>No Mapping</i>
Description	<i>&lt;Home directory&gt;</i>

<b>Field Name</b>	<b>Value</b>
Parameter Name	<i>Shell</i>
Type	<i>STRING</i>
Default Value	<i>None</i>
Is the parameter required?	Yes
Is the parameter encrypted?	No
Display Name	<i>Login Shell</i>
Mapping	<i>No Mapping</i>
Description	<i>&lt;Login Shell&gt;</i>

### *Command Code*

<b>Field Name</b>	<b>Value</b>
-------------------	--------------

Shell Command	usermod -d \${Home} -s \${Shell} \${Account} 2>&1
---------------	---

### 3. Reset account's password

While defining the "Reset an Account's password" verb, certain parameters need to be defined. The following tables will help you define these parameters. Each table is dedicated to one parameter.

#### Command Input Parameters

Field Name	Value
Parameter Name	<i>Account</i>
Type	<i>STRING</i>
Default Value	<i>None</i>
Is the parameter required?	Yes
Is the parameter encrypted?	No
Display Name	<i>Account Name</i>
Mapping	<i>\${Account.Name}</i>
Description	<i>&lt;Name of the account&gt;</i>

Field Name	Value
Parameter Name	<i>Password</i>
Type	<i>STRING</i>
Default Value	<i>None</i>
Is the parameter required?	Yes

Is the parameter encrypted?	Yes
Display Name	<i>Password</i>
Mapping	<i>No Mapping</i>
Description	<i>&lt;Initial password to reset to&gt;</i>

### Command Code

Field Name	Value
Shell Command	<code>echo \${Password}   passwd \${Account} --stdin 2&gt;&amp;1</code>

### 4. Add an account to a group

While defining the “Add an Account to group” verb, certain parameters need to be defined. The following tables will help you define these parameters. Each table is dedicated to one parameter.

### Command Input Parameters

Field Name	Value
Parameter Name	<i>Account</i>
Type	<i>STRING</i>
Default Value	<i>None</i>
Is the parameter required?	Yes
Is the parameter encrypted?	No
Display Name	<i>Account Name</i>
Mapping	<i>\${Account.Name}</i>

Description	<Name of the account>
-------------	-----------------------

### Command Code

Field Name	Value
Shell Command	<code>gpasswd -a \${Account} \${Group} 2&gt;&amp;1</code>

### 5. Remove an account from a group

While defining the “Remove an Account from group” verb, certain parameters need to be defined. The following tables will help you define these parameters. Each table is dedicated to one parameter.

### Command Input Parameters

Field Name	Value
Parameter Name	<i>Account</i>
Type	<i>STRING</i>
Default Value	<i>None</i>
Is the parameter required?	Yes
Is the parameter encrypted?	No
Display Name	<i>Account Name</i>
Mapping	<i>\${Account.Name}</i>
Description	<Name of the account>

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

Parameter Name	<i>Group</i>
Type	<i>STRING</i>
Default Value	<i>None</i>
Is the parameter required?	Yes
Is the parameter encrypted?	No
Display Name	<i>Group Name</i>
Mapping	<i>\${Group.Name}</i>
Description	<i>&lt;Name of the group&gt;</i>

### Command Code

Field Name	Value
Shell Command	<i>gpasswd -d \${Account} \${Group} 2&gt;&amp;1</i>

### 6. Enable an account

While defining the “Enable an Account” verb, certain parameters need to be defined. The following tables will help you define these parameters. Each table is dedicated to one parameter.

### Command Input Parameters

Field Name	Value
Parameter Name	<i>Account</i>
Type	<i>STRING</i>
Default Value	<i>None</i>

Is the parameter required?	Yes
Is the parameter encrypted?	No
Display Name	<i>Account Name</i>
Mapping	<i>\${Account.Name}</i>
Description	<i>&lt;Name of the account&gt;</i>

### Command Code

Field Name	Value
Shell Command	<i>passwd -u \${Account} 2&gt;&amp;1</i>

### 7. Disable an account

While defining the “Disable an Account” verb, certain parameters need to be defined. The following tables will help you define these parameters. Each table is dedicated to one parameter.

### Command Input Parameters

Field Name	Value
Parameter Name	<i>Account</i>
Type	<i>STRING</i>
Default Value	<i>None</i>
Is the parameter required?	Yes
Is the parameter encrypted?	No
Display Name	<i>Account Name</i>

Mapping	<code>\${Account.Name}</code>
Description	<code>&lt;Name of the account&gt;</code>

### Command Code

Field Name	Value
Shell Command	<code>passwd -l \${Account} 2&gt;&amp;1</code>

### 8. Create a group

While defining the “Create a Group” verb, certain parameters need to be defined. The following tables will help you define these parameters. Each table is dedicated to one parameter.

### Command Input Parameters

Field Name	Value
Parameter Name	<code>Group</code>
Type	<code>STRING</code>
Default Value	<code>None</code>
Is the parameter required?	Yes
Is the parameter encrypted?	No
Display Name	<code>Group Name</code>
Mapping	<code>\${Group.Name}</code>
Description	<code>&lt;Name of the group&gt;</code>

### Command Code

Field Name	Value
Shell Command	<code>groupadd \${Group} 2&gt;&amp;1</code>

### 9. Delete a group

While defining the “delete a Group” verb, certain parameters need to be defined. The following tables will help you define these parameters. Each table is dedicated to one parameter.

### Command Input Parameters

Field Name	Value
Parameter Name	<i>Group</i>
Type	<i>STRING</i>
Default Value	<i>None</i>
Is the parameter required?	Yes
Is the parameter encrypted?	No
Display Name	<i>Group Name</i>
Mapping	<code>\${Group.Name}</code>
Description	<i>&lt;Name of the group&gt;</i>

### Command Code

Field Name	Value
Shell Command	<code>groupdel \${Group} 2&gt;&amp;1</code>



## Copyrights

Copyright © 2017 Dell Inc. or its subsidiaries. All Rights Reserved.

## Trademarks

Dell, RSA, the RSA Logo, EMC and other trademarks, are trademarks of Dell Inc. or its subsidiaries. Other trademarks may be trademarks of their respective owners. For a list of RSA trademarks, go to [www.emc.com/legal/emc-corporation-trademarks.htm#rsa](http://www.emc.com/legal/emc-corporation-trademarks.htm#rsa).