Advanced Authentication Card Utility

Micro Focus Advanced Authentication provides a REST API that is used by this utility to register authentication cards and associate them with users.

The utility is implemented in Java and require JRE 1.8 or higher. The components of the utility are:

* dist/AAFUtility.jar – contain the utility
* /lib/commons-codec-1.11.jar – The Apache codec library
* /lib/json-20180813.jar – The org.json library
* AAFUtility.properties – The properties file used to configure the utility.

The utility requires a CSV file as input. The format of the file is:

Repository\userID,site ID,cardID,password

For example:

LOCAL\admin,184,271212,password01

If all the users are in a single repository then you may omit the “Repository\” if you specify the userRepository property in the properties file. Note that you must include two slashes after the repository name in the properties file.

Before the utility can be used you must create an Endpoint definition for the REST API. This is done in the AAF Admin interface.

1. Select Endpoints
2. Select Add

A screenshot of a cell phone

Description automatically generated

1. Set the name as “RESTAPI”
2. Set the description to “Used by AAFUtility”
3. Set “Is Enabled” and “Is Trusted” to ON
4. Click Save. The Endpoint ID and Endpoint Secret will be displayed.

A screenshot of a cell phone

Description automatically generated

1. Copy the Endpoint ID and Endpoint Secret to the properties file.

NOTE: If you fail to capture the secret you will need to delete and recreate the Endpoint

There are four other properties that need to be set in the properties file:

* The adminID property is set to the account that will execute the commands. This account must have admin rights to authenticate to the Helpdesk UI. The account must also have a password set for the password method. The property must include the repository and must have four slashes. For example:

LOCAL\\\\admin

* The mode property must be set to one of three values: add, update, clear
  + “add” will create a card entry for each entry in the file where the user does not already have a card
  + “update” will set the users card to the card entry in the file regardless of any pre-existing card
  + “clear” will remove the card from each user entry in the file.
* The type property must be set to one of three values: card, password, both
  + “card” will ignore the password field and only modify the card registered for the user
  + “password” will ignore the card and site fields and will only modify the password credential of the user
  + “both” will modify both the card and the password
* The targetURL property is set to the base URL of the AAF server. For example:

<https://aaf61.pointbluetech.com>

NOTE!! Please ensure that there are no trailing spaces on the lines of the property file.

Once the properties file is complete, the utility run from the command prompt by changing to the “dist” directory and executing the command:

java -jar AAFUtility.jar <path to config file> <path to data file> <admin password>

For example, if the properties file and data file are in the dist directory:

java -jar AAFUtility.jar AAFUtility.properties badgeData.csv myPassword

The utility produces a log file each time it is run that list the result for each entry in the data file.