

# Credentials Manager for ZebraTester

## Function

Credentials Manager is an [Apica ZebraTester plugin](#) that extends the functionality of secure storage and retrieval of credentials or passwords using Apica's ASM API.

- It's using, but is not limited to, a combination of [AES/ECB/PKCS5Padding](#) and [RSA 512bit Key Pair](#) as its current encryption algorithm.

**i** The transformation [AES/ECB/PKCS5Padding](#) tells the `getInstance` method to instantiate the `Cipher` object as an [AES](#) cipher with ECB mode of operation and PKCS5 padding scheme.

- It can support different types of encryption algorithms.

After retrieving the encrypted username/password from API, the plugin will decrypt data to be used in the ZebraTester script.

To store the username/password to the Apica ASM servers, run the JAR plugin from the command line and pass the necessary parameters as we'll describe below.

**✓** This JAR plugin supports batch processing as well as encrypting/decrypting passwords.

## Use Cases

Use Credentials Manager when:

- You are doing ASM monitoring and
- You need to pass username/password (or any key/value combinations) in the script but do not want the text string to be stored and transmitted in cleartext.

**⚠** This plug-in is not intended for Load Testing scenarios.

## CyberArk or Credentials Manager?

Some companies use **CyberArk** to store and fetch their credentials. But if not using a CyberArk Enterprise Password Vault for this purpose, the Credentials Manager can be used to establish login monitoring, with **test** monitoring usernames and passwords to reasonably obfuscate the passwords.

## Components

- ZebraTester V5.5F or later
- CredentialsManager2UtilityTool\_vXX.jar
- CredentialsManager2.class
- [Apica Synthetic Monitoring API](#)
  - A Custom Dictionary is where this information is sent to/from.

Method	Route	Description
POST	<a href="#">/scenarios/proxysniffer/dictionaries</a>	Adds the ZebraTester scenario custom dictionary.
GET	<a href="#">/scenarios/proxysniffer/dictionaries/{dictionary_key}</a>	Gets a ZebraTester scenario custom dictionary by <b>dictionary key</b> . A custom dictionary can contain any data used by Proxy Sniffer scripts that need to be stored separately from scripts.
PUT	<a href="#">/scenarios/proxysniffer/dictionaries/{dictionary_key}</a>	Replaces ZebraTester scenario custom dictionary. <b>NOTE</b> that this will replace any existing data already in the dictionary!
DELETE	<a href="#">/scenarios/proxysniffer/dictionaries/{dictionary_key}</a>	Delete a Proxy Sniffer scenario custom dictionary by dictionary key.

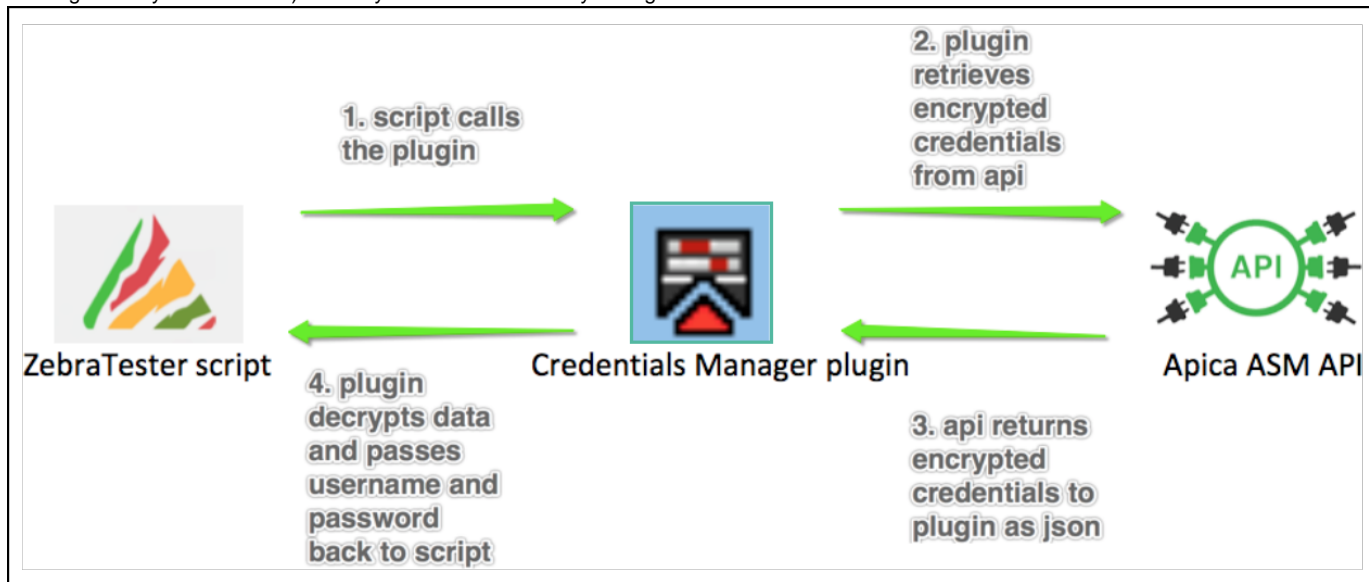
 Downloads:

The files you'll need are available here:

- [Credentials Manager 2 Utility Tool jar file](#)
- [Credentials Manager 2 class file](#)

Credentials Manager Process Flow:

During ZebraTester script execution, this plugin will run and retrieve the encrypted credentials using Apica's ASM API, which is encrypted at rest and during transport. The plugin will decrypt the credentials using the specified encryption algorithm and preset, unique, symmetric key (assigned or designated by the customer) and only stored in the memory during the session.



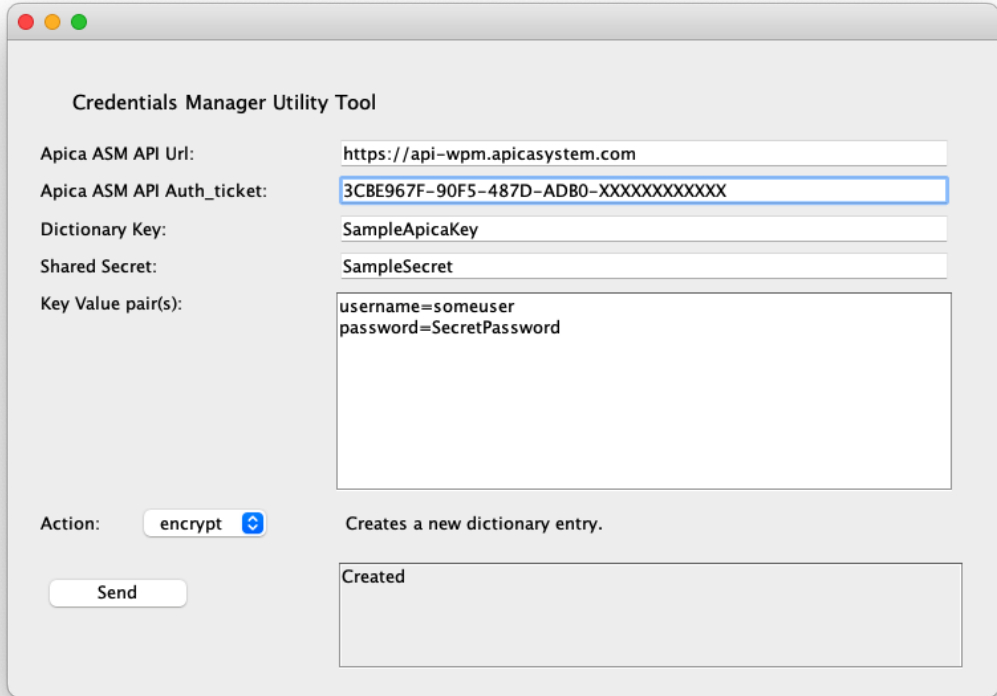
Installing the Credentials Manager Plugin

Step	Information
------	-------------

**1. Mandatory:**  
**Create an Entry (a new custom dictionary)**

First, create the encrypted credentials entry into Apica's ASM API by running the jar version of the plugin from a command line to open GUI.

```
java -jar CredentialsManager2UtilityTool_vXX.jar
```



Key:Value pair(s) will be created as a result.

**⚠** Up to 50 key:value pairs can be used for a single dictionary\_key as long as the value doesn't exceed 1000 characters. Credentials Manager 2 also supports long values over 1000 characters (e.g., private keys or certs) but will reduce the number of keys:value pairs that can be stored to around 5 entries.

**2. Add the Credentials Manager plugin into ZebraTester**

By adding the .class as a plugin to your ZebraTester Script and then passing in the ASM\_API\_Url, ASM\_API\_AuthTicket, dictionary\_key, shared\_secret, and key as an index parameter, you can fetch the encrypted value, which the plugin will decrypt at execution rather than having that value hard-coded into the script as plaintext.

**Usage & Syntax:**

To retrieve encrypted credentials and decrypt to be used in the ZebraTester script.

Input Parameter			
Number	Name	Mandatory?	Assign From
1	Apica ASM API URL	Yes	variable
2	Apica ASM API Auth Ticket	Yes	variable
3	Dictionary Key	Yes	variable

4	Shared Secret Key	Yes	variable
5	Key (of value)	Yes	variable
<b>Output Parameter</b>			
<b>Number</b>	<b>Name</b>	<b>Mandatory?</b>	<b>Extract To</b>
1	Value (of Key)	Yes	[variable]

CredentialsManager2UtilityTool\_vXX.jar (Used from the GUI)

Func tion	Set-up Instructions
<b>Encry pt</b> key /value pairs like User ame and Passw ord	<div style="border: 1px solid #ccc; padding: 10px; margin-bottom: 10px;"> </div> <ul style="list-style-type: none"> <li>• A successful execution of this will result in Created</li> <li>• To verify it would be using the API command with the above username as the dictionary_key as an example:</li> <li>• <code>https://api-wpm.apicasystem.com/v3/scenarios/proxysniffer/dictionaries/SampleApicaKey?auth_ticket=[authticket]</code> <ul style="list-style-type: none"> <li>• It would return the json entry of the encrypted value(s) for the key, SampleApicaKey</li> </ul> </li> </ul> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> </div>

Update  
key  
/value  
pairs  
like  
User  
ame  
and  
Passw  
ord

**Credentials Manager Utility Tool**

Apica ASM API Url:

Apica ASM API Auth\_ticket:

Dictionary Key:

Shared Secret:

Key Value pair(s):

Action:  Append key/value pairs to existing dictionary entries.

```
{  
  "password": "ptaPB8wNB1NeqhBji7I9+S5JrcQ6XnPL3+sjr99qkX1c8HvHchS8iHU1XJb8aKnOWPfkOahhD/8oepEGRT1FA==",  
  "username": "J1tvGEfYCD5nr8c2xH0ro9mN9Bt/txF/8ALFy8CgNUKe5C5Snogujp8zoxFonFduxKYz/Qw2YB9W88M2vxV85Q=="  
}
```

Overwrite key value pairs like Username and Password

**Credentials Manager Utility Tool**

Apica ASM API Url:

Apica ASM API Auth\_ticket:

Dictionary Key:

Shared Secret:

Key Value pair(s):

Action:   Store/overwrite data. Creates a new entry or overwrites existing ones.

Decrypt/Retrieve keys like Password

**Credentials Manager Utility Tool**

Apica ASM API Url:

Apica ASM API Auth\_ticket:

Dictionary Key:

Shared Secret:

Key Value pair(s):


Action:   Retrieve and decrypt existing api entry.

Example:

1. CD to a Java /bin location (I used ZebraTester 7B's java /bin directory)
2. Download current version of "Credentials Manager" jar (as of writing it is CredentialsManager2UtilityTool\_v06.jar) and move to the /bin location you CD'ed to
3. Run this command:

```
java -jar CredentialsManager2UtilityTool_vXX.jar
```

Note that Apica ASM API Url (<https://api-wpm.apicasystem.com>) is the base URL only (no path like v3/scenarios/proxysniffer/dictionaries/)

 More information about ZebraTester Plugins can be found in the <https://apica-kb.atlassian.net/wiki/spaces/DAZT/pages/148865368/ZebraTester+Plug-in+Handbooks#%22Load-Test-Plug-In%22-Developer-Handbook>