



# **Bot Runner Password Management Readme**

---

**Version 1.0  
25/06/2020**

# Table of Contents

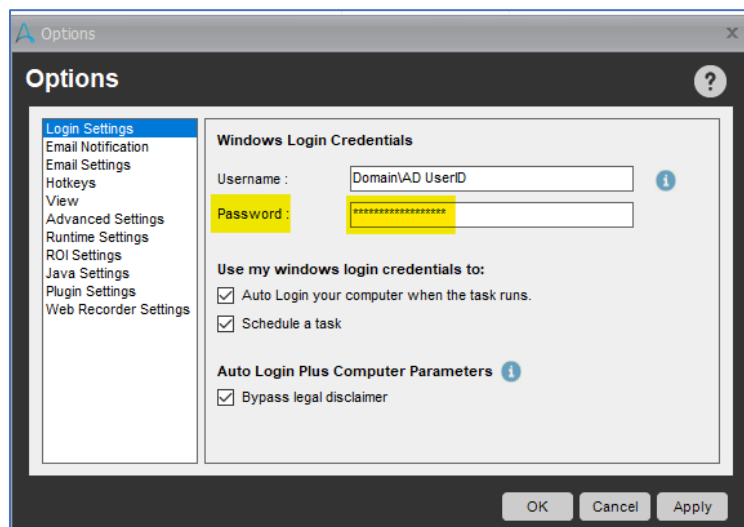
<b>1. Introduction.....</b>	<b>3</b>
1.1 Overview .....	3
1.2 Common Use cases.....	10
<b>2. Requirements &amp; Prerequisites .....</b>	<b>11</b>
2.1 System Requirements.....	11
2.2 Prerequisites .....	11
2.3 Security Measures .....	12
2.4 Disclaimers.....	12
<b>3. Getting Started .....</b>	<b>14</b>
3.1 Skill Matrix .....	14
3.2 Installation Hierarchy.....	15
3.3 Quick Start .....	15
3.3.1 Setup.....	16
3.3.2 Configuration .....	16
<b>4. Reports .....</b>	<b>18</b>
<b>5. Logs.....</b>	<b>19</b>
<b>6. Troubleshooting &amp; Support.....</b>	<b>21</b>
6.1 Support.....	21
6.2 FAQs.....	<b>Error! Bookmark not defined.</b>
<b>Appendix A: Record of Changes.....</b>	<b>22</b>
<b>Appendix B: Acronyms.....</b>	<b>23</b>
<b>Appendix C: References .....</b>	<b>24</b>

## 1. Introduction

This document contains all essential information for the user to make full use of the Bot or Digital worker. This manual includes a description of the functions and capabilities and step-by-step procedures for setup & configuration of the Bot.

### 1.1 Overview

At a high level this process accomplishes the task of resetting a bot runner's Active Directory password and pushing that updated password into said Bot Runners Client Login settings. Thereby allowing a cyclical password reset process per your organizations requirements, minimizing human interaction and human error. Having the passwords reset in an automated fashion will have a tremendous effect on any audits due to the consistency and accuracy of the automated process every time.



- First, deploy from the Control Room or run - **MainTask\_BotPasswordRotation.atmx**.
- **ReadConfig\_BotPasswordRotation.atmx**
  - Collect Day, Month, Year, Hour, Minute and Second system variables and cast them to user created vDay, vMonth, etc variables.

\$	Variable Operation: \$Month\$ To \$vMonth\$
\$	Variable Operation: \$Day\$ To \$vDay\$
\$	Variable Operation: \$Year\$ To \$vYear\$
\$	Variable Operation: \$Hour\$ To \$vHour\$
\$	Variable Operation: \$Minute\$ To \$vMinute\$
\$	Variable Operation: \$Second\$ To \$vSecond\$

- Set vMonth/vDay/vYear to mProcess\_Start\_Date variable.
  - We are using 'm' for Metrics related variables.

```

$ Variable Operation: "$vMonth$/$vDay$/$vYear$" To $mProcess_Start_Date$
$ Comment: ***** REMOVING QUOTES SO DATA CAN BE WRITTEN AS A STRING IN METRICS FILE *****
$ String Operation: Replace "" with "" in "$mProcess_Start_Date$" and assign output to $mProcess_Start_Date$
$ Comment:

```

- Cast vConfig\_File\_Path with Configuration File location so it can be read.
- Loop through csv and assign Configuration Variables
  - Create Metrics and Run Log Filenames

```

If $Filedata Column(1)$ Equal To (=) "cRun_Log" Then
  Comment: Please enter the conditional commands here.
  Variable Operation: $Filedata Column(2)$ To $cRun_Log$
  Comment: ***** ESTABLISHING METRICS AND RUNLOG FILENAMES *****
  Variable Operation: $cRun_Log$\ADM_Pass_Rotation-$vMonth$-$vDay$-$vYear$-$vHour$-$vMinute$-$vSecond$_RunLog.csv To $cRun_Log$
  Comment:
Else If $Filedata Column(1)$ Equal To (=) "cMetrics_Log" Then
  Comment: Please enter the conditional commands here.
  Variable Operation: $Filedata Column(2)$ To $cMetrics_Log$
  Comment: ***** ESTABLISHING METRICS AND RUNLOG FILENAMES *****
  Variable Operation: $cMetrics_Log$\ADM_Pass_Rotation-$vMonth$-$vDay$-$vYear$-$vHour$-$vMinute$-$vSecond$_Metrics.csv To $cMetrics_Log$

```

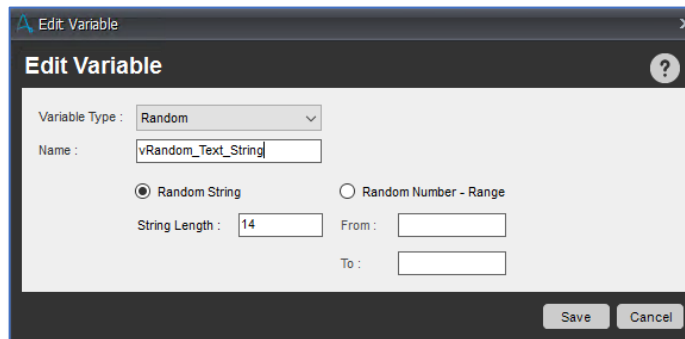
- Create headers for Run Log and Metrics Files

```

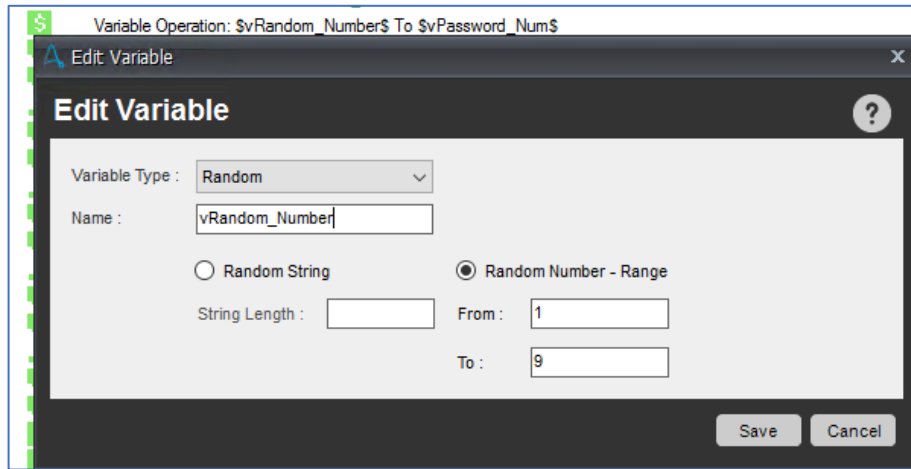
$ Comment: ***** CREATE HEADERS FOR RUNLOG AND METRICS FILES *****
$ Log to File: Process Start Time,Machine Name,Username,Process Name,Line Item Start,Line Item End,Line Item ID,Status,Exception Description in "$cMetrics_Log$"
$ Log to File: Timestamp,Machine Name,Username,Process Name,Task Name,Message,Log Message,Line Number in "$cRun_Log$"

```

- **SetPasswordAndPushToActiveDirectory.atmx**
  - vRandom\_Text\_String creates a random set of 14 characters in Variable Manager and assigns that string to vBeginning\_Random\_Password.



- vRandom\_Number pulls a random number between 1 and 9 then assigns it to vPassword\_Number.



- String Operation to cut the first 7 characters from vBeginning\_Random\_Password and assigns it to vPassword\_Caps
- String Operation to cut the last 7 characters and assign to vPassword\_Lower\_Case
- String Operation to capitalize all letters in vPassword\_Caps
- String Operation to set all letters in vPassword\_Lower\_Case to lower case.
- Assemble and assign modified caps variables and lower case plus the number at the end. Creating a vNew\_Password of 7 upper case and 7 lower case letters plus a number. 15 total characters per ADM's password policy

Variable Operation: \$vPassword\_Caps\$\$vPassword\_Lower\_Case\$\$vPassword\_Num\$ To \$vNew\_Password\$

- Executes Active Directory command to send new password to Active Directory for identified User Name (no domain is needed for Modifying identified username). Using Test in below example.
  - NOTE: System Variable \$Username\$ can be in **User Name** field when doing real world testing and moving into production. The whole goal of his process is to reset the password for the Username on which the process is currently running – allowing for Username system variable to be set. In my below example we parsed out to allow for testing against different usernames than the machine on which testing was occurring.

Active Directory

?

Select Command:

☐ Create User
☐ Create Group
☐ Create Object
☐ Search

☒ Modify User
☐ Modify Group
☐ Modify Object
☐ Get Property

Enter LDAP Path:

Parent Path: \$cLDAP\_Path\$

...

e.g. LDAP://DomainServer/CN=Users,DC=Domain,DC=Com

Login User: \$cReset\_Bot\_Username\$

Login Password: \*\*\*\*

Select User Action:

Select Action: Change Password

▼

Enter Modify User Details:

User Name: \$vControl\_Room\_Username\$

New Password: \*\*\*\*

Confirm Password: \*\*\*\*

💡 TIP: Press F2 to insert variable.

Save

Cancel

- **WritePasswordToLoginSettings.atmx**
  - Execute REST Web Service command to produce an API Authorization Key.
    - We only need a username with Admin rights in Control Room as well as "Generate API-Key" role
    - The API Key Generated MUST be generated by the same individual whose username we are using.

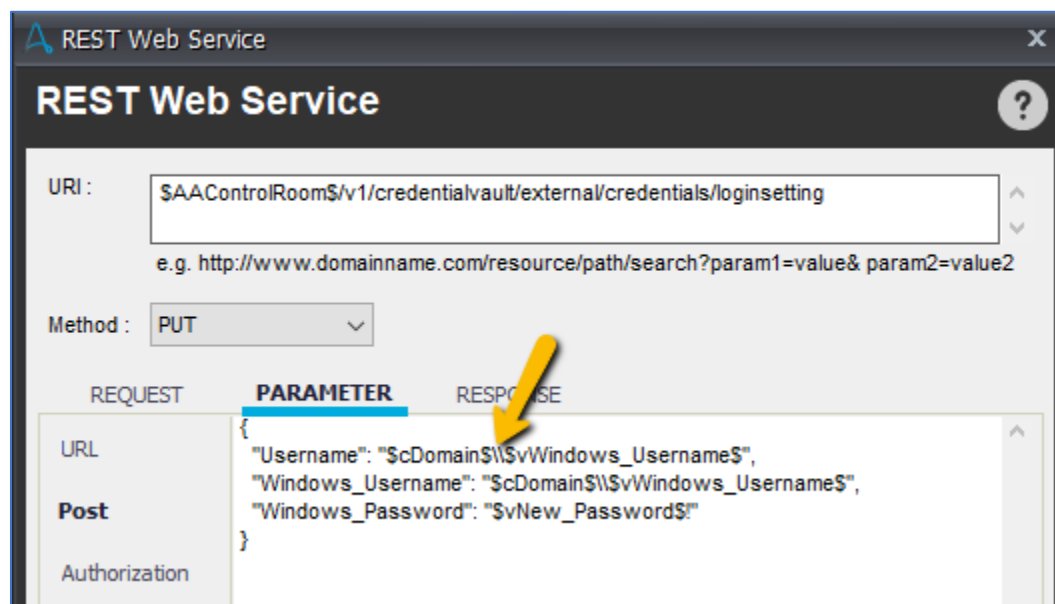


```

IF
  If $vAuthToken$ Not Equal To (<>) "" Then
    Comment: Please enter the conditional commands here.
    Variable Operation: Auth Token Successfully obtained. Token is $vAuthToken$ To $vLog_Message$
    Log to File: ,Machine$,System(USERNAME)$,$vProcess_Name$,SAATaskName$,vLog_Message$,Info, in "$cRun_Log$"
  Else
    Variable Operation: System Exception To $mItem_Status$
    Variable Operation: API Auth Token could not be obtained. vAuthToken currently equals $vAuthToken$. To $vException_Description$
    Variable Operation: Could not be obtained. vAuthToken currently equals $vAuthToken$. To $vLog_Message$
    Log to File: ,Machine$,System(USERNAME)$,$vProcess_Name$,SAATaskName$,vLog_Message$,Error, in "$cRun_Log$"
    Stop The Current Task
  End If

```

- If it is null – Throwing Exception to Main Task, sending email and writing as an exception.
- If not = null, Execute Put REST Web Service command to write password to designated username Client Login Settings.
- Please note on the below screenshot the double backslashes between Domain and Username. This is REQUIRED for the REST command to properly find your credentials. A single backslash will result in an error.



- Process returns to **MainTask\_BotPasswordRotation.atmx**
  - Logs status to Metrics file
  - Sends email to cRPA\_Ops\_Email with Metrics file attached.

**Send Email** ?

**Select Email Parameters :**

From :

\$cFrom\_Email\$

To :

\$cRPA\_Ops\_Email\$

Cc:

Bcc:

Subject:

\$vProcess\_Name\$ Has Finished

Attachment:

\$cMetrics\_Log\$<File Size Not Available>;

^

⏏

Ⓢ

☒ Validate if attachment is missing

Message :

☒ Text ☐ HTML

Please contact \$cRPA\_Ops\_Email\$ for any issues.

^

⏏

## 1.2 Common Use cases

The beauty of this bot is that it is an Automation Anywhere focused utility so anyone who wants to consistently update their Bot Runner Active Directory passwords without the hassle of having to login to each machine can use it.

---

## 2. Requirements & Prerequisites

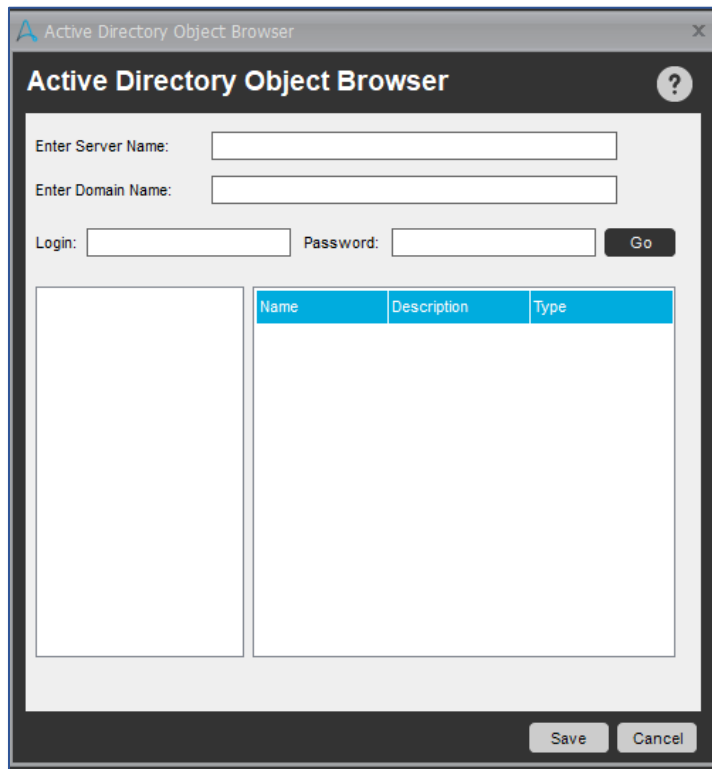
---

### 2.1 System Requirements

- For the PC or server where the bot needs to run:
  - RAM: 8GB or higher
  - PROCESSOR: Intel Core i5 or higher and equivalent for any other OS
  - Hard Disk: Up to 2GB of overall free space in the AA Client installation drive.
  - Reference below for Enterprise Client & Control Room system requirements.
    - <https://docs.automationanywhere.com/bundle/enterprise-v11.3/page/enterprise/topics/aae-architecture-implementation/deployment-planning/requirements-server-hardware.html>

### 2.2 Prerequisites

- Automation Anywhere Enterprise – v11.3.2 and later
  - In order to connect to API you will need a user's credentials with:
    - Administrative access
    - Generate API-Key role:  
<https://docs.automationanywhere.com/bundle/enterprise-v11.3/page/enterprise/topics/control-room/control-room-api/authenticate-apikey.html>
    - <https://docs.automationanywhere.com/bundle/enterprise-v11.3/page/enterprise/topics/control-room/control-room-api/control-room-apikey-role.html>
- Active Directory Prerequisites:
  - LDAP Server Name
  - LDAP Domain Name
  - You will need to identify and grant a specific username permission access LDAP Server and to set passwords for your identified Bot Runner usernames. "Login User" fields in below screenshot.
  - **\*\*Note**, when validating the required Active Directory data points, it is recommended to fill out the fields showing in the below screenshot and hit 'Go'. If successful the main window will populate with data and once you click 'Save', this information automatically populates into appropriate fields.
    - Once the final, full LDAP path is populated you can save it to *cLDAP\_Path* within the configuration file.



- Initial login to Bot Runner Client Login Settings:
  - This process is setup with the assumption that the Login Settings on the target client machine have already been set at least once. Hence why we are using a 'PUT' method rather than a 'POST' method.
  - If the Client has never had login settings filled out yet, you would a) login once to establish the credentials or b) change the method to POST.

## 2.3 Security Measures

Consider how an RPA team representative will be made aware of what the updated AD password is. Preferably adding a module to log the updated password to a secure location. In the event of manually accessing the Bot Runner, an end user may need to be aware of the current password. This component was intentionally left out as what is considered 'secure' is potentially different for every organization.

## 2.4 Disclaimers

- *Please note that this process will simply accomplish the task of tying together the components of Active Directory connection and interacting with Automation Anywhere's*

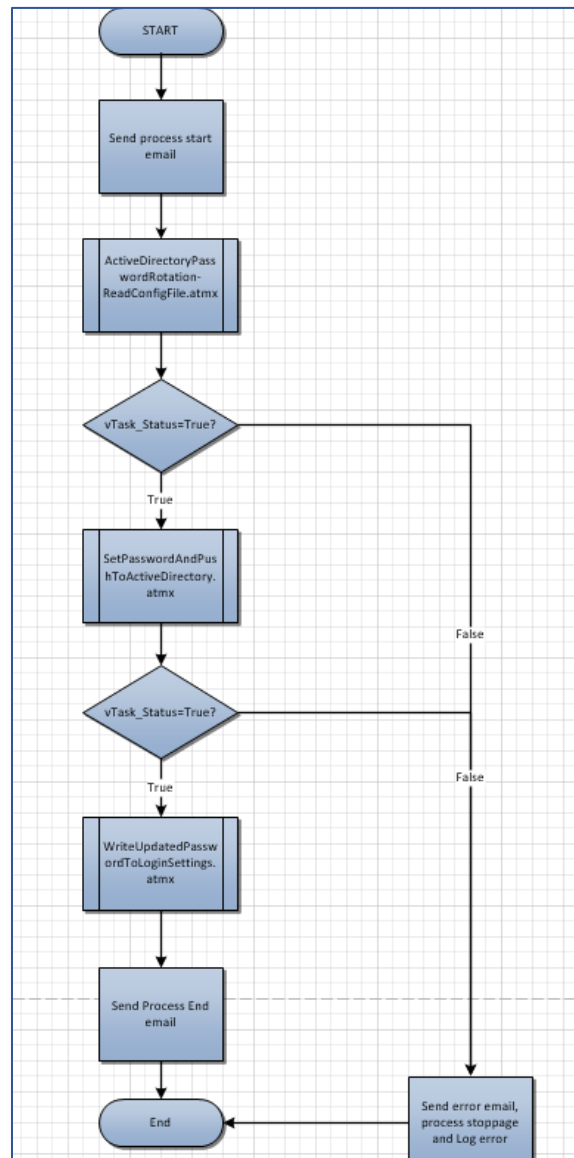


---

*API calls. How you choose to store or relay your password once changed is up to your organizational requirements and may require an expansion to this template.*

## 3. Getting Started

### 3.1 Skill Matrix



## 3.2 Installation Hierarchy

Once the bot is downloaded and installed, the installer creates the files in the folder structure as shown below.

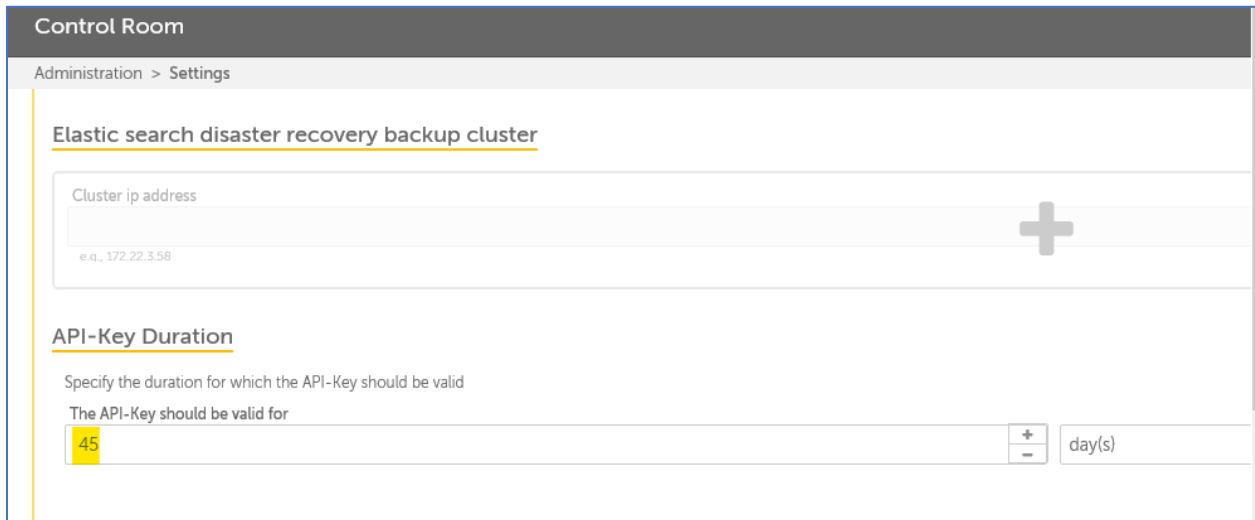
Folder Structure	Description
<b>&lt;AAApplication Path&gt;</b>  <b>1. My Tasks</b>  <b>1.1. Bot Store</b>  <b>1.1.1. ActiveDirectoryPasswordRotation-Agilify</b>  <ul style="list-style-type: none"> <li>• <b>Error Folder</b> <ul style="list-style-type: none"> <li>○ <b>Logs Folder</b></li> <li>○ <b>Snapshots Folder</b></li> </ul> </li> <li>• <b>Input Folder</b> <ul style="list-style-type: none"> <li>○ HRWorkerConfiguration.txt</li> </ul> </li> <li>• <b>My Tasks (all .atmx files)</b> <ul style="list-style-type: none"> <li>○ ActiveDirectoryPasswordRotation-MainTask</li> <li>○ ActiveDirectoryPasswordRotation-ReadConfigFile</li> <li>○ SetPasswordAndPushToActiveDirectory</li> <li>○ WriteUpdatedPasswordToLoginSettings</li> </ul> </li> </ul>	<b>&lt;AA Application Path&gt;</b> is the location where AA files are stored on your machine.  <b>1. My Tasks:</b> My Task Folder is the default directory where Bot files are saved.  <b>1.1. Bot Store:</b> Bot Store folder contains the Bot Name folder which the installer creates while installing the Bot.  <b>1.1.1. ActiveDirectoryPasswordRotation-Agilify:</b> This folder gets created by the installer and contains the essential configuration file.  <ul style="list-style-type: none"> <li>• <b>Error Folder:</b> This is where logs immediately will be stored once installing this package. Can be adjusted in configuration file per your organizational requirements for logging</li> <li>• <b>Input Folder:</b> Where files being fed into process is saved. Currently the only input is process Configuration file. <ul style="list-style-type: none"> <li>○ BotPasswordRotation_ConfigFile.csv: This is the first item requiring to be filled out prior to being able to successfully run the Bot.</li> </ul> </li> <li>• <b>My Tasks (all .atmx files):</b> Folder containing all the developed Platform Source Code.</li> </ul>

## 3.3 Quick Start

### 3.3.1 Setup

#### Control Room API Key Creation:

- Please follow the attached steps:
  - <https://docs.automationanywhere.com/bundle/enterprise-v11.3/page/enterprise/topics/control-room/control-room-api/control-room-apikey-role.html>
  - Note: You can extend the API Key validity duration up to 45 days (it is 1 day by default).
    - While logged in as an AAE\_Admin, go to Administration => Settings => Control Room Database & Software tab
    - Click Edit
    - Update API-Key Duration Field and Save changes



### 3.3.2 Configuration

Provide configurable variable list with the details as mentioned in the example

For External File –

Variable Name	Type	Mandatory	Purpose
cRun_Log	Text/Number	Yes	Folder where we write Run log files to
cMetrics_Log	Text/Number	Yes	Folder where we are writing Metrics files to
cLDAP_Path	Text/Number	Yes	Full LDAP Path once configured with Server and Domain Name
cReset_Bot_Username	Text/Number	Yes	This is the identified username which has rights to change other Bot Runners in



			<i>Active Directory</i>
cAPI_Key_Generator_Username	<i>Text/Number</i>	<i>Yes</i>	<i>Admin username which also has access to user created API Key Generator Role</i>
cAPI_Key	<i>Text/Number</i>	<i>Yes</i>	<i>Actual 40 character API Key</i>
cRPA_Ops_Email	<i>Text/Number</i>	<i>Yes</i>	<i>Email address where critical emails will be sent for direct action.</i>
cFrom_Email	<i>Text/Number</i>	<i>Yes</i>	<i>Email address from which emails will be 'sent'</i>
cDomain	<i>Text/Number</i>	<i>Yes</i>	<i>Organization's domain when logging into Automation Anywhere. Same domain as is input when connecting to Control Room.</i>

---

## 4. Reports

---

*No Bot Insight Reports are associated with this process.*

## 5. Logs

The only Logs associated with this process are a Run Log which documents general process steps and a Metrics Log which summarizes whether the 'work item' was successful. Once the package is downloaded the path will automatically to

<\$AAApplicationPath\$\Automation Anywhere\My Tasks\Bot Store\ActiveDirectoryPasswordRotation-Agilify\Error Folder\Logs>

Which is meant to be a placeholder. The final location can be updated in your configuration file associated with this process.

### Current Packaged State:

1	Config Variable Name:	Config Variable Value:
2	cRun_Log	C:\Users\<Username>\Documents\Automation Anywhere Files\Automation Anywhere\My Tasks\Bot Store\ActiveDirectoryPasswordRotation-Agilify\Error Folder\Logs
3	cMetrics_Log	C:\Users\<Username>\Documents\Automation Anywhere Files\Automation Anywhere\My Tasks\Bot Store\ActiveDirectoryPasswordRotation-Agilify\Error Folder\Logs

When read by the Read Config File task, it will evaluate if the process is writing logs locally and if so, replace all text up to Automation Anywhere Files with \$AAApplicationPath\$ variable as mentioned above.

### In Configuration File:

```

30 IF $Filedata Column(1)$ Equal To (=) "cRun_Log" Then
31     Comment: Please enter the conditional commands here.
32     $ Variable Operation: $Filedata Column(2)$ To $cRun_Log$
33     IF $cRun_Log$ Includes "C:\Users\<Username>\Documents\Automation Anywhere Files" Then
34         Comment: Replacing specific machine path with the more portable system variable AAApplicationPath
35         String Operation: Replace "C:\Users\<Username>\Documents\Automation Anywhere Files" with "$AAApplicationPath$" in "$cRun_Log$" and assign output to $cRun_Log$
36     End If
37     Comment: ***** ESTABLISHING METRICS AND RUNLOG FILENAMES *****
38     $ Variable Operation: $cRun_Log$\AD_Pass_Rotation-$vMonth$_$vDay$_$vYear$_$vHour$_$vMinute$_$vSecond$_RunLog.csv To $cRun_Log$
39     Comment:



```

Filename naming conventions are:

**Run Log:** \$cRun\_Log\$\AD\_Pass\_Rotation-\$vMonth\$\_\$vDay\$\_\$vYear\$\_\$vHour\$\_\$vMinute\$\_\$vSecond\$\_RunLog.csv

**Metrics Log:** \$cMetrics\_Log\$\AD\_Pass\_Rotation-\$vMonth\$\_\$vDay\$\_\$vYear\$\_\$vHour\$\_\$vMinute\$\_\$vSecond\$\_Metrics.csv

Ex:

Documents > Automation Anywhere Files > Automation Anywhere > My Tasks > Bot Store > ActiveDirectoryPasswordRotation-Agilify > Error Folder > Logs			
Name	Date modified	Type	Size
 AD_Pass_Rotation-6_26_2020-8_6_46_Metrics	6/26/2020 8:07 AM	Microsoft Excel C...	1 KB
 AD_Pass_Rotation-6_26_2020-8_6_46_RunLog	6/26/2020 8:07 AM	Microsoft Excel C...	1 KB

*Note that we created the logs to write independently – this is because some organizations may want the day-to-day operations logs (Run Logs) written to a different location than the 'Work Item' logs or the Metrics Logs.*

---

## 6. Troubleshooting & Support

---

### 6.1 Support

Please Contact Von Miles at Agilify and state you are requesting support for Bot Runner Password Management process from the AA Bot Store.

Email: [von.miles@agilifyautomation.com](mailto:von.miles@agilifyautomation.com)

Phone: [317-343-7028](tel:317-343-7028)

---

## Appendix A: Record of Changes

*Instructions: Provide information on the version number, the date of the version, the author/owner of the version, and a brief description of the reason for creating the revised version.*

Version Number	Date of Change	Author	Notes
<b>1.0</b>	6/26/2020	Matthew Newell	Initial document creation

---

## Appendix B: Acronyms

*Instructions: Provide a list of acronyms and relative description used within the document. Sequence should be alphabetical*

Acronym	Description
<b>AA</b>	<b>Automation Anywhere</b>
<b>AD</b>	<b>Active Directory</b>
<b>API</b>	<b>Application programming interface</b>
<b>LDAP</b>	<b>Lightweight Directory Access Protocol</b>
<b>VM</b>	<b>Virtual Machine</b>

## Appendix C: References

*Instructions: Summarize the relationship of this document to other relevant references. Provide information for all documents/links used to arrive at and/or referenced within this document.*

*Example:*

No.	Topic	Reference Link
1	Virtual Machine Specs	<a href="https://docs.automationanywhere.com/bundle/enterprise-v11.3/page/enterprise/topics/aae-architecture-implementation/deployment-planning/requirements-server-hardware.html">https://docs.automationanywhere.com/bundle/enterprise-v11.3/page/enterprise/topics/aae-architecture-implementation/deployment-planning/requirements-server-hardware.html</a>
2	<i>Using API Key for Credential Vault External API</i>	<a href="https://docs.automationanywhere.com/bundle/enterprise-v11.3/page/enterprise/topics/control-room/control-room-api/authenticate-apikey.html">https://docs.automationanywhere.com/bundle/enterprise-v11.3/page/enterprise/topics/control-room/control-room-api/authenticate-apikey.html</a>
3	<i>Creating API Key Generator Role</i>	<a href="https://docs.automationanywhere.com/bundle/enterprise-v11.3/page/enterprise/topics/control-room/control-room-api/control-room-apikey-role.html">https://docs.automationanywhere.com/bundle/enterprise-v11.3/page/enterprise/topics/control-room/control-room-api/control-room-apikey-role.html</a>