Control Room APIs

The Automation Anywhere Enterprise Control Room provides various public APIs which allow you to customize your business automation for third party applications. It enables the third party applications to consume RPA, orchestrate bots and manage the RPA data based on events.

Control Room offers the following APIs:

1. Authentication API
2. Bot Login Credentials API
3. Bot Deployment API
4. Export Import Bot API

API for authentication

All the APIs are preceded by an Authentication API, wherein the user invoking third party application has to authenticate so as to use the downstream APIs. Click here for details.

API to manage bot login credentials

In enterprise organizations where the password rotation policy is applied, the Client user has to remember to update the password from Tools → Options → Login Settings during each password rotation. To automate this process, Control Room enables you to use an API to create, update or delete the Login Credentials stored in the credential vault. Click here for details.

API for bot deployment

To deploy Bots onto the Automation Environment the user has to login into Control Room, select the Bot and the Bot Runners and then 'Run/Schedule' the task. However, as the Automation scenarios scale up, there is an increasing need to deploy/trigger Bots from an external third party application. To meet this business requirement, you can use the Bot Deployment API which enables you to trigger a Bot from an External System. Click here for details.

API for export import of bots for Business Lifecycle Management

Usually, the Control Room user has to depend on means other than Control Room (for example email) to deploy TaskBots from one environment to another. Using the Export-Import APIs, you can easily introduce a customized BLM solution thus removing all external factors that could possibly disrupt your automation life cycle. Click here for details.

Control Room API for Authentication

Control Room v.11 exposes public APIs so as to manipulate the Control Room data and to deploy bots from an external system. This enables third party applications to consume RPA, orchestrate bots and to manage the RPA data based on events - for example, create an AAE user as soon as a user is created in SAP system; update the login credentials in AAE as soon as password is rotated in domain controller.

All the APIs are preceded by an Authentication API, wherein the invoking third party application user has to authenticate so as to use the downstream APIs.

Authentication API

API: v1/authentication

If the Control Room URL is https://ultron.com:81, then the API will be https://ultron.com:81/v1/authentication

The API takes two parameters as input in JSON format:

1. The username of the AAE user
2. The password of the AAE user

```
Note: All parameters are mandatory.
```

For example:

1. The AAE username is **mike_williams**.
2. The AAE password is **abc123**, then the JSON will be :

```
 {
"Username":"mike_williams",
”Password":"abc123"
 }
```

If the authentication is successful, the Control Room will issue an authentication token which needs to be passed on to the Deployment API as header information.

```
Note: The authentication token will ONLY be valid for 15 minutes from the time it is issued.
```

Authentication API Response Codes

Http Status code	Response	Description

200	{
	"token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.
	eyJ1c2VyIjoiMSIsImlzcyI6Imh0dHA6Ly9sb2NhbGhvc3Qvd2ViY3
	JzdmMvIiwiYXVkIjoiaHR0cDovL2xvY2FsaG9zdC93ZWJjcnN2Yy8iLCJle
	HAiOjE0OTUwOTAwOTksIm5iZiI6MTQ5NTA4OTE5OX0.qPPhpti0j7
	LGAmWkj3XFymFfJXzA1P4zPehIjVYfuIc"
	}
		Authentication is successful

401	{
	"message": "Invalid credentials."
	}
		○ The password is invalid
		○ User does not exist
		○ AD Authentication - Credentials are Invalid

401	{ "message": "Please verify your email by clicking on the email verification link. This is mandatory as you will be able to login post verification only."	Email notification is enabled - user has not verified email

	}	

402	{
	"message": "License expired."
	}
		License has expired

403	{
	"message": "Your account is not activated. Please contact the admin."
	}
		User is deactivated

Control Room API for Bot Deployment

To deploy Bots onto the Automation Environment, currently, the user has to login into Control Room, select the Bot and the Bot Runners and then 'Run/Schedule' the task.

However, as the Automation scenarios scale up, there is an increasing need to deploy/trigger Bots from an external third party application.

To meet this business requirement, Automation Anywhere Enterprise has published Application Programming Interfaces (APIs) using which a Bot can be triggered from an External System.

A Control Room user can use these APIs to deploy Bots (Tasks) to BotRunners on commencement of events specified by a third party/external application.

Key Features and Business Benefits of Control Room APIs

- Bots can be deployed from an external third party systems using AAE APIs.
- The input and output of APIs is JSON based (industry standard data-interchange format).
- Bot Deployment can be orchestrated from an External Application / Workflow using a combination of scripts and AAE APIs.

Note: The Bot deployment API can ONLY be invoked once the system/user has authenticated using the Authentication API

Also, the user will need to have the 'Run my bots' privileges and the privileges of the bot runners on which the bot is to be deployed.

Deployment API

The Deployment API is used to deploy Bots to BotRunners.

API: <Control Room URL>/v1/schedule/automations/deploy

For example, if the Control Room URL is https://www.ultron.com:81 ; then the bot deployment API will be https://www.ultron.com:81/v1/schedule/automations/deploy

The user can pass three parameters as JSON string.

1. Bot name with relative path - This is mandatory.
2. List of Bot-Runners and users in JSON format - This is mandatory.
3. Use RDP based approach - This is optional and set to **false** by default.

Deployment Scenario and corresponding JSON string:

1. For example, the name of the Bot is AccountsBot.atmx and the Bot is under 'My Tasks'
2. The Bot is to be deployed on 3 machines
    - First machine hostname BR-1 with user U-1
    - Second machine hostname BR-2 with user U-2
    - Third machine hostname BR-3 with user U3.
3. The JSON string in the above scenario will be:

```
 {
"taskRelativePath":"My Tasks\\AccountsBot.atmx", "botRunners":
[
{
"client":"BR-1", "user":"U-1"
},
{
"client":"BR-2", "user":"U-2"
},
{
"client":"BR-3", "user":"U-3"
}
]
}
```

Bot Deployment API Response Codes

Http(s) Status code	Response - Description
200	Succesful creation of automaton.
400	Bad Request
401	Authentication Required
403	Unauthorized access
409	Conflict
500	Internal Server error

Control Room API to manage bots login credentials

When the Bot is deployed from Control Room to the Bot Runner, the Bot will auto-login into the Bot Runner (if the machine is locked / logged off). The Bot will use the credentials stored in the Credential Vault for auto-login. These credentials are set by the user using the Tools → Options → Login Settings of AAE Client.

However, there could be cases when the user's Windows password is modified; especially in Enterprises where there is a password rotation policy. In such cases, the user has to remember to update the new password from Tools →Options → Login Settings.

In order to automate the above process, Control Room 11.1 provides a direct API to create, update or delete the Login Credentials stored in the credential vault.

There are 2 steps to use this API:

1. Invoke the Authentication API
2. Invoke the Login Credentials API

Authentication API

The details for the Authentication API are provided in the article Control Room API for Authentication.

Login Credentials API

Note that only the Control Room Administrators (users having AAE_Admin Role) can use the Login Credentials API; this means that if the user invoking the Authentication API has an AAE_Admin role, only that user can use the Login Credentials API.

Also, this API will make use of the authentication token that is obtained using the Authenication API. The authentication token has to be passed on as one of the header inputs to the Login Credentials API.

API: v1/credentialvault/external/credentials/loginsetting

If the Control Room URL is https://ultron.com:81, then the API will be: https://ultron.com:81/v1/credentialvault/external/credentials/loginsetting

The API takes three parameters as input in JSON format:

1. The username of the AAE user.
2. The Login (Windows) username for the user which is to be updated in the Credential Vault against that user.
3. The Login (Windows) password for the user which is to be updated in the Credential Vault against that user.

```
Note: All parameters are mandatory.
```

For example:

If we take a 'Database Authentication' scenario (where users are stored and authenticated by the Control Room) and

1. The AAE username is **mike_williams**
2. The fully qualified Login username is **ultron.com\mike.williams**
3. The Login password is **abc123**, then the JSON will be
   { "Username": "mike_williams", "Windows_Username": "ultron.com\\mike.williams", "Windows_Password": "abc123" }

Using the Login Credentials API, the Control Room Admins can:

1. Create a user's Login Credentials in the Control Room - by using the 'POST' method
2. Update a user's Login Credentials in the Control Room - by using the 'PUT' method
3. Delete a user's Login Credentials from Control Room - by using the 'DELETE' method.

VB Script to Create User's Login Credentials

```
'AUTHENTICATION API - START
Set objStream = CreateObject("ADODB.Stream")
objStream.CharSet = "utf-8"
objStream.Open
objStream.LoadFromFile("D:\Deven.Deshpande\Office\Products\API for CV\auth-input.txt")

restRequest = objStream.ReadText()

objStream.Close
Set objStream = Nothing

contentType = "application/json"

Set oWinHttp = CreateObject("WinHttp.WinHttpRequest.5.1")

oWinHttp.Open "POST", "http://productslt12.aaspl-brd.com:81/v1/authentication", False

oWinHttp.setRequestHeader "Content-Type", contentType

oWinHttp.Send restRequest

response = oWinHttp.StatusText

MsgBox response

Dim AuthToken

'Set oJson = new aspJSON

AuthToken = oWinHttp.ResponseText

'oJson.loadJSON(AuthToken)

'MsgBox oJson.data("token")

MsgBox AuthToken

'AUTHENTICATION API - ENDS

'---------------------------------------------------

'RESPONSE HEADER PARSING - START
Dim sToken
Dim posUser

sToken = Right(AuthToken, Len(AuthToken) - 10)

MsgBox(sToken)

'sToken = Left(sToken, Len(sToken) - 4)

posUser = InStr(sToken, "user")

MsgBox posUser

sToken = Left(sToken, posUser - 4)

MsgBox(sToken)

'RESPONSE HEADER PARSING - END

'DEPLOYMENT API - START

Set objStream_dep = CreateObject("ADODB.Stream")
objStream_dep.CharSet = "utf-8"
objStream_dep.Open
objStream_dep.LoadFromFile("D:\Deven.Deshpande\Office\Products\API for CV\cred-input.txt")

restRequest = objStream_dep.ReadText()
```

```
objStream_dep.Close
Set objStream = Nothing

contentType = "application/json"

Set oWinHttp = CreateObject("WinHttp.WinHttpRequest.5.1")

oWinHttp.Open "POST", "http://productslt12.aaspl-brd.-
com:81/v1/credentialvault/external/credentials/loginsetting", False

oWinHttp.setRequestHeader "Content-Type", contentType

oWinHttp.setRequestHeader "X-Authorization", sToken

oWinHttp.Send restRequest

response = oWinHttp.StatusText

MsgBox response

Dim DeployResponse

DeployResponse = oWinHttp.ResponseText

'MsgBox DeployResponse

'DEPLOYMENT API - ENDS
```

Contents of Input Files

Applicable to the above VB Script

The text Of auth-input.txt will have the input JSON String For authentication **{"UserName":"admin","Password":"12345678"}**

The text For deploy-input.txt will have the input JSON String For deployment
{ "Username": "mike_williams", "Windows_Username": "ultron.com\\mike.williams", "Windows_Password": "abc123" }

API - Export and Import Bots for Bot Lifecycle Management

As a Control Room user with privileges to Export bots and files as well as Import bots and files, you can export and then import TaskBots with its dependent files in different automation environments to help manage your organization's Bot Lifecycle Management (BLM).

You can use the Control Room Export Import REST API to manage your automation TaskBots including dependent files in different environments such as Development, Testing, Acceptance, and Production based on your organization's automation needs.

Usually, the Control Room user has to depend on means other than Control Room (for example email) to deploy TaskBots from one environment to another. Using the Export-Import APIs, you can easily introduce a customized BLM solution thus removing all external factors that could possibly disrupt your automation life cycle.

Refer Export bot files and Import bot files to use the functionality from your Control Room user interface.

Features and benefits

- Flexibility to customize the process life-cycle environment (Bot Lifecycle Management) and not restrict it to the Development, Testing, Acceptance and Production environments.
- Role based access control on Bot Lifecycle Management
- Automatic export of dependencies (files and bots)
- Audit and traceability on source and target environment for compliance
- Notification references in email on successful execution or failure of export and import.

Prerequisites

Export

- The Control Room user whose credentials are used for authentication must have "Export bots" permission
- The Control Room user whose credentials are used for authentication must have download permission on the Bots and dependencies that are being exported.
- If Version Control is enabled in the source Control Room, the production version of all bots and dependencies which you want to export must be set.
- User account that is used to run the Control Room services must have access to the location where package is getting exported, e.g. network location(shared drive) or on Control Room server machine

Import

- The Control Room user whose credentials are used for authentication must have "Import bots" permission
- The Control Room user whose credentials are used for authentication must have upload permission on the Bots and dependencies that are being imported.
- The Control Room user who will execute the utility to import multiple bots must have access to the exported package file provided by Automation Anywhere.

API Endpoints

a. **Export -** <Control Room URL>/v1/alm/export

b. **Import -** <Control Room URL>/v1/alm/import

Using the above end points of the BLM Export API you can export a single bot and all of its dependencies.

Export Bot

Export a single bot with its dependent files using the Export API provided by Automation Anywhere:

1. Use the Post method to generate a token using the end point **http(s)://<hostname:port>/v1/authentication**. For this provide the Control Room instance as **Server Name /Hostname /IP** and the **Port** number. Refer sample:
```
https://crdevenv.com:81/v1/authentication
```
2. Use the Post method and state the parameters for credentials in Body Data. Refer sample:
```
{
"username":"cradmin",
"password":"cr@admin"
}
```

3. Click Play/Start
4. Once your credentials are authenticated, use the generated token for Export API.
5. Provide parameters such as **filename**, **destination path** and **package name** in Body Data to export a bot. Following list provides parameter description:
    - **filename** - use this to specify the filepath ending with the name of the file that has to be expored.
    - **destination path** - use this to specify the destination path where the exported package is to be stored
    - **package name** - use this to specify the package name that you must provide

      Refer sample,

      ```
      {
       "filePath":"Automation Anywhere\\My Tasks\\Finance\\Account Reconciliation.atmx",
       "destinationPath":"tempshare\\datashare\\Finance Department",
       "packageName":"Finance package"
       }
      ```
6. Click Play/Start
7. You can use multiple sources to view the export results in,
    - Reponse Data that comprise **package path** and **checksum**.
    - Audit Log page (Main as well as details)
    - Email when you receive notification on success or failure

Import Bot

Once the bot is successfully exported to a network drive or Control Room machine path, another user can import that package to a different Control Room using the Import API:

1. Use the Post method to generate a token using the end point **http(s)://<hostname:port>/v1/authentication**. Provide the Control Room instance as **Server Name** /**Hostname** /**IP** and the **Port** number. Refer sample:

   ```
   https://crtestenv.com:82/v1/authentication
   ```
2. Use the Post method and state the parameters for credentials in Body Data.

   Refer sample:

   ```
   {
   "username":"cradmin2",
   "password":"cr@admin"
   }
   ```
3. Click Play/Start
4. Once your credentials are authenticated, provide parameters such as **package path** and the **checksum** that was generated as a token during export in Body Data.Refer sample:

   ```
   {
   "packagePath":"tempshare\\datashare\\Finance Department\\Finance Package_20171221-154403.aapkg",
   "checksum":"ZLyQ+Lbu2N+beEuXf6qd2Qi9uwi3BZxApn57C7mYjKQ="
   }
   ```

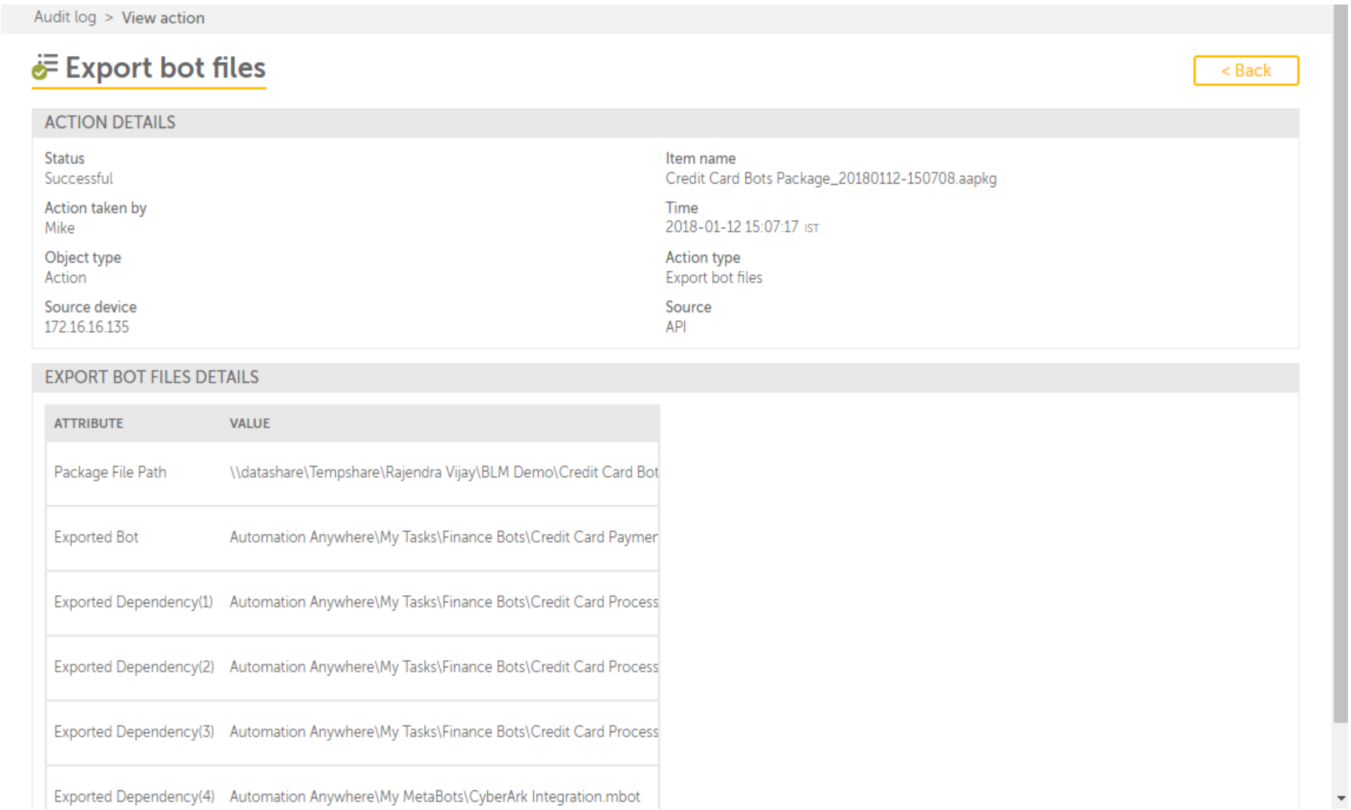
API Response Codes

Http(s) Status code	Response - Description
200	Package created successfully
400	Bad request parameter
404	File not found
501	Permission error

Audit Logs

An audit entry is logged in the Control Room **Audit Log** page when you export or import bots. The illustration below shows the detailed audit entries for Export Bots and Import Bots

Export bots audit details



Export bot files

ACTION DETAILS

Status
Successful

Item name
Credit Card Bots Package_20180112-150708.aapkg

Action taken by
Mike

Time
2018-01-12 15:07:17 IST

Object type
Action

Action type
Export bot files

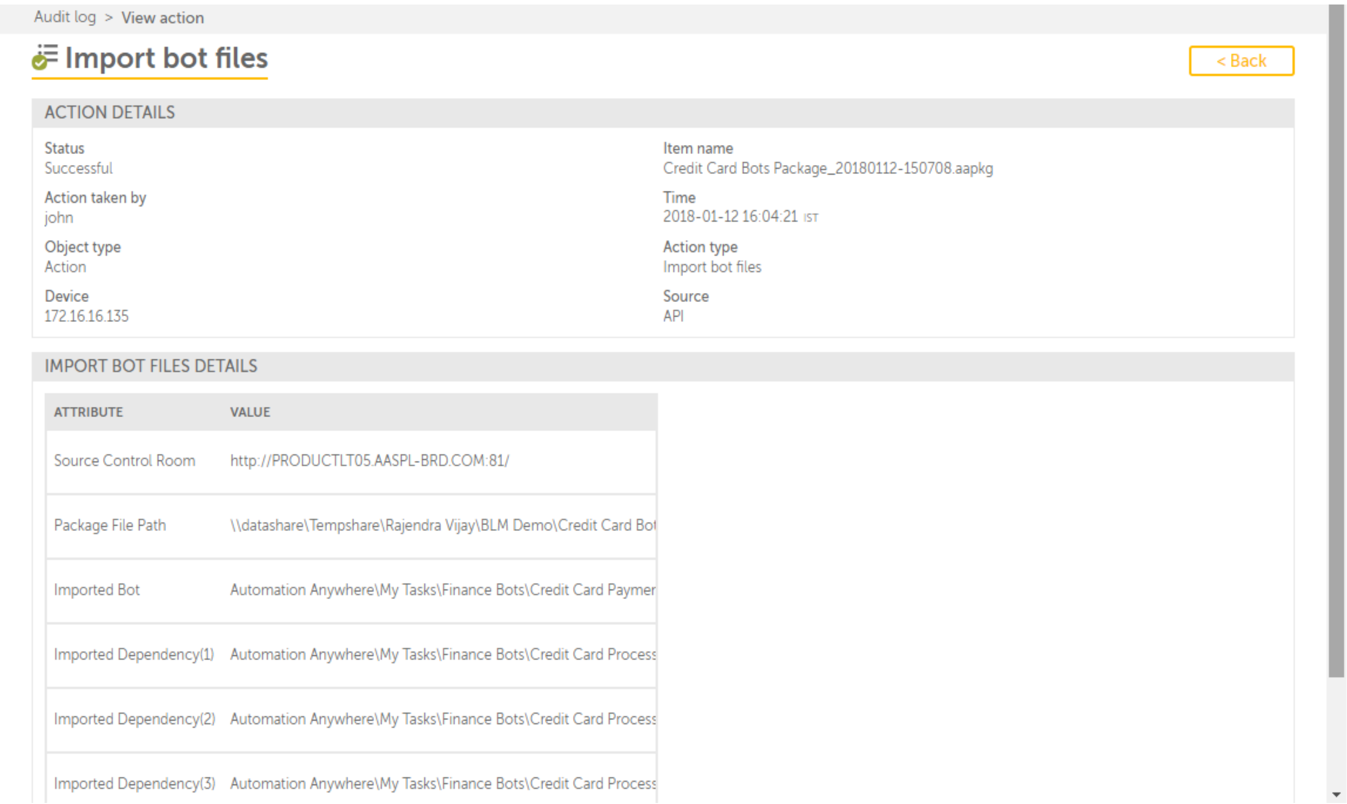
Source device
172.16.16.135

Source
API

EXPORT BOT FILES DETAILS

ATTRIBUTE	VALUE
Package File Path	\\datashare\Tempshare\Rajendra Vijay\BLM Demo\Credit Card Bot
Exported Bot	Automation Anywhere\My Tasks\Finance Bots\Credit Card Paymer
Exported Dependency(1)	Automation Anywhere\My Tasks\Finance Bots\Credit Card Process
Exported Dependency(2)	Automation Anywhere\My Tasks\Finance Bots\Credit Card Process
Exported Dependency(3)	Automation Anywhere\My Tasks\Finance Bots\Credit Card Process
Exported Dependency(4)	Automation Anywhere\My MetaBots\CyberArk Integration.mbot

Import bots audit details



ACTION DETAILS

Status
Successful

Item name
Credit Card Bots Package_20180112-150708.aapkg

Action taken by
john

Time
2018-01-12 16:04:21 IST

Object type
Action

Action type
Import bot files

Device
172.16.16.135

Source
API

IMPORT BOT FILES DETAILS

ATTRIBUTE	VALUE
Source Control Room	http://PRODUCTLT05.AASPL-BRD.COM:81/
Package File Path	\\datashare\Tempshare\Rajendra Vijay\BLM Demo\Credit Card Bot
Imported Bot	Automation Anywhere\My Tasks\Finance Bots\Credit Card Paymer
Imported Dependency(1)	Automation Anywhere\My Tasks\Finance Bots\Credit Card Process
Imported Dependency(2)	Automation Anywhere\My Tasks\Finance Bots\Credit Card Process
Imported Dependency(3)	Automation Anywhere\My Tasks\Finance Bots\Credit Card Process