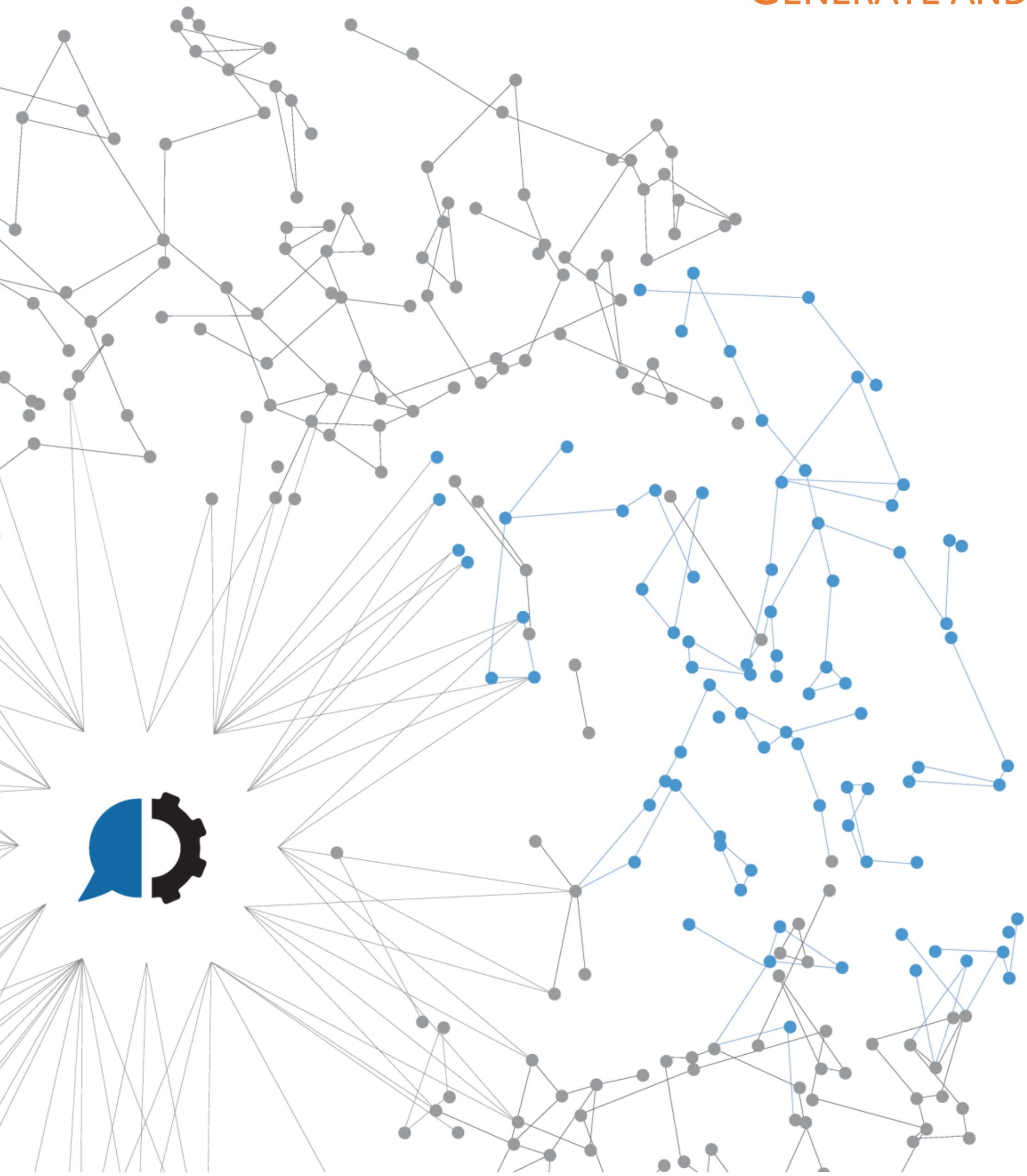
Adobe Sign Bot

Generate and Refresh Token



Content

Overview

This how to guide introduces the *Adobe Sign Bot - Generate and Refresh Token*. Instead of spending precious time and resources on repetitive, manual tasks, let Automate go to work and streamline your IT and business processes with robotic process automation. Automate can take over the manual steps which greatly reduces repetitive activities and improves the quality and consistency of the work. The top processes being automated are report generation, file movement, data import and export, and scheduling batch processing.

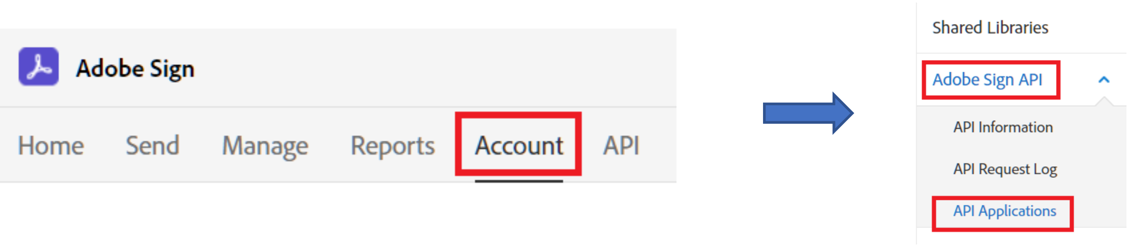
The ***Adobe Sign Bot - Generate and Refresh Token*** is based in the ***Adobe Sign OAuth Authorization Code Workflow***. It executes the **OAuth2** access token request ***"/oauth/token"*** to the URL provided by Adobe Sign. It uses a *"grant_type"* parameter with ***"authorization_code"*** (authorization code workflow) type for first time token generation. Once the token is generated within the first execution of the bot, it will use a *"grant_type"* with ***"refresh_token"*** in order to refresh the token during next executions. The expiration time for the token is by default 1 hour. We recommend to analyze its behavior and set the execution of this bot accordingly to your needs in order to use the generate token on a daily basis and make any Adobe Sign Bot work with it.

Is important to note that ***Adobe Sign only supports APIs Integration with its "Enterprise Plan".***

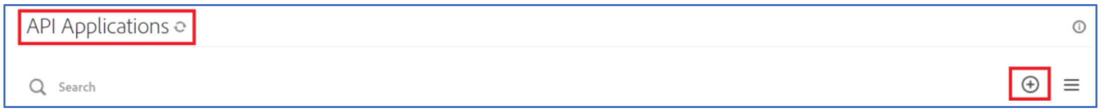
Carefully check the PREREQUISITES and the HOW TO EXECUTE THE ADOBE SIGN BOT - GENERATE AND REFRESH TOKEN sections to get detailed information.

PREREQUISITES

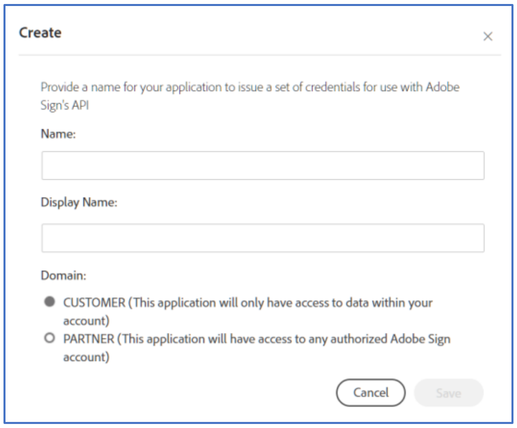
- Automate: The Adobe Sign Bot depends on Automate software in order to work. The minimal supported versions are:
    - Automate Ultimate 11.2
    - Automate Plus 11.2
    - Automate Desktop 11.2
- Automate Markup Language file (.AML): The primary file type used in Automate which contains the steps of our Adobe Sign Bot task
- Adobe Sign Graph: Our minimal requirements are
    - Create an application in Adobe Sign on the Application List Page: You will need to create an application within the Adobe Sign platform to be able to configure the proper scopes and requests for the token that will be used for the connections of other Adobe Sign Bots. You will find it by signing into your Adobe Sign account going to the ***ACCOUNT*** tab > **Adobe Sign API** > ***API Applications*** section. You will need to be an administrator, so if you do not have permissions, ask your Adobe Sign Administrator. More information on how to create the App can be find in this Adobe Sign Link.



Add an API Application:



Complete the Name, Display Name and select the Domain according to your needs:



Create

Provide a name for your application to issue a set of credentials for use with Adobe Sign's API

Name:

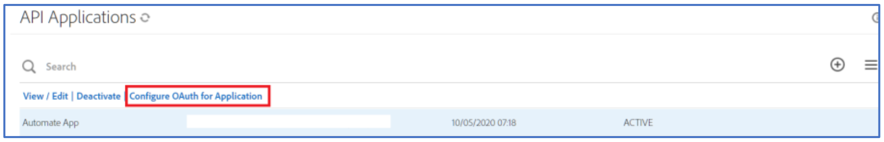
Display Name:

Domain:

● CUSTOMER (This application will only have access to data within your account)

○ PARTNER (This application will have access to any authorized Adobe Sign account)

- Configure OAuth for your application: Once you created your Application, it will appear in the API Applications list. Select yours and click on ***"Configure OAuth for Application"***. A configuration form will be opened. Complete it according to your needs. Within this configuration you will find the ***Client ID*** and ***Client Secret*** that will be set in the Bot Configuration (Keep them safely). You will find more information about the Authorization Code Grant Workflow here.

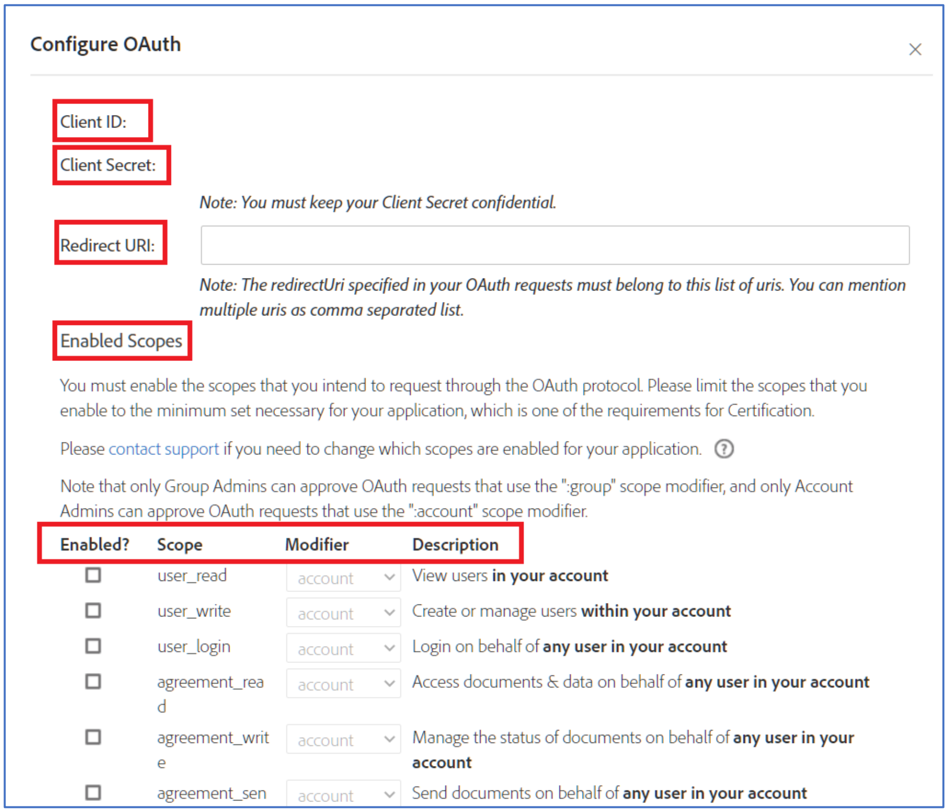


API Applications

View / Edit | Deactivate | Configure OAuth for Application

Automate App 10/05/2020 07:18 ACTIVE

OAuth config: Set the ***Redirect URL*** (you can use *https://localhost*) and select the Scopes that you will need:



Configure OAuth

Client ID:

Client Secret:

Note: You must keep your Client Secret confidential.

Redirect URI:

Note: The redirectUri specified in your OAuth requests must belong to this list of uris. You can mention multiple uris as comma separated list.

Enabled Scopes

You must enable the scopes that you intend to request through the OAuth protocol. Please limit the scopes that you enable to the minimum set necessary for your application, which is one of the requirements for Certification.

Please contact support if you need to change which scopes are enabled for your application.

Note that only Group Admins can approve OAuth requests that use the ":group" scope modifier, and only Account Admins can approve OAuth requests that use the ":account" scope modifier.

Enabled?	Scope	Modifier	Description
□	user_read	account	View users **in your account**
□	user_write	account	Create or manage users **within your account**
□	user_login	account	Login on behalf of **any user in your account**
□	agreement_read	account	Access documents & data on behalf of **any user in your account**
□	agreement_write	account	Manage the status of documents on behalf of **any user in your account**
□	agreement_sen	account	Send documents on behalf of **any user in your account**

- Authorization Code for first bot execution: Once the application is created and configured you will be able to get the Authorization Code needed for first time bot execution. You will need to open in your browser a personalized URL (change the ***##CLIENT_ID##*** with yours) and you will receive the Authorization code in your browser address bar. Note that in the URL we must specify the **scopes** needed and the redirect URI. In this example the scopes are: ***user_login:self, agreement_send:account and agreement_write:account***. Some other scopes might be added if needed. You will need to provide your login information and once you accept the request permission, you'll get the Authorization Code within your browser address bar.

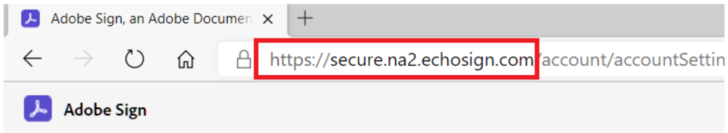
  Remember that the code is associated to the username provide when login and accepting the access request, as well as the token generated with the Auth Code.

Next you can find an example of the request URL, the Acceptance Window, and how you get the code:

Request URL:

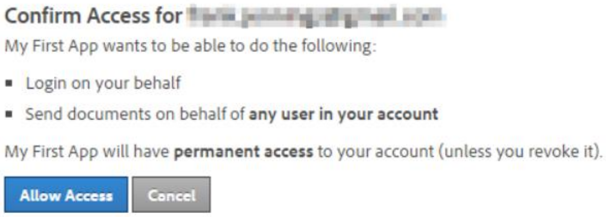
[***https://secure.na2.echosign.com****/public/oauth?redirect_uri=****https://localhost****&response_type=code&client_id=****##CLIENT_ID##****&scope=user_login:self+agreement_send:account+agreement_write:account*](https://secure.na2.echosign.com/public/oauth?redirect_uri=https://localhost&response_type=code&client_id=##CLIENT_ID##&scope=user_login:self+agreement_send:account+agreement_write:account)

Important*: The Adobe Sign URL (https://secure.na2.echosign.com/ in last example) will be the same that you get when signing into Adobe Sign with your account. You can copy it from your browser address bar:*



Adobe Sign, an Adobe Documen
https://secure.na2.echosign.com/account/accountSettin
Adobe Sign

Acceptance Window:



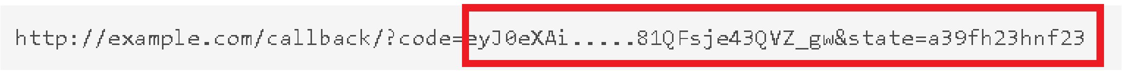
Confirm Access for [redacted]

My First App wants to be able to do the following:

- Login on your behalf
- Send documents on behalf of **any user in your account**

My First App will have **permanent access** to your account (unless you revoke it).

Authorization Code:



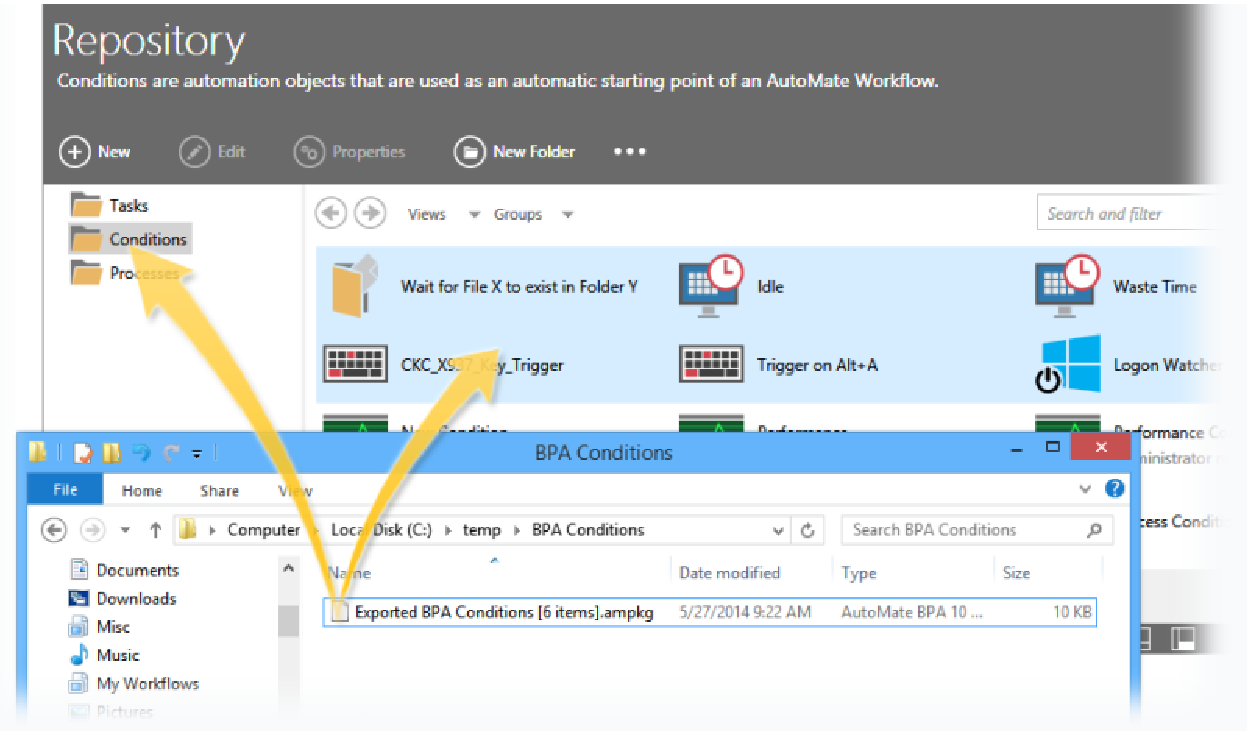
```
http://example.com/callback/?code=eyJ0eXAi.....81QFsje43QVZ_gw&state=a39fh23hnf23
```

How to import an Automate task

Compatible file types can be imported to the repository via drag-and-drop

From the Server Management Console, navigate to the Repository section

Drag the desired file(s) from its original location and drop them into the folder in the SMC. Files can be dropped into the folder icon or the main panel (as shown below). Imported object(s) are automatically placed into their corresponding repository location.



Repository

Conditions are automation objects that are used as an automatic starting point of an AutoMate Workflow.

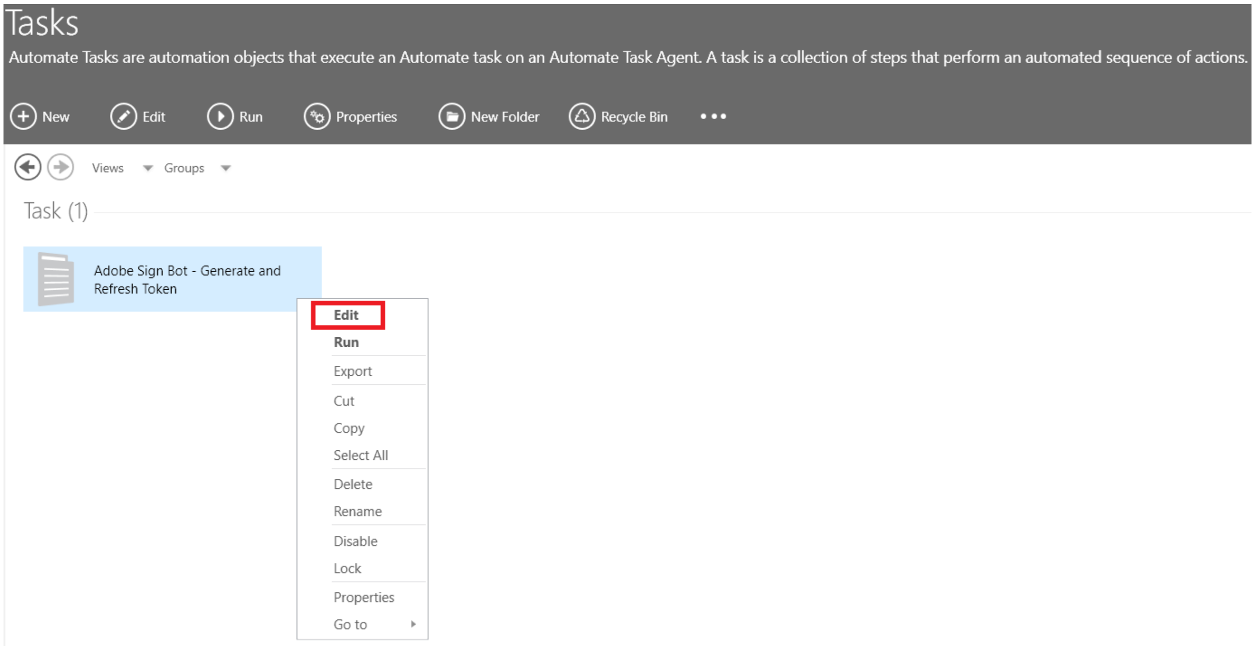
Name	Date modified	Type	Size
Exported BPA Conditions [6 items].ampkg	5/27/2014 9:22 AM	AutoMate BPA 10 ...	10 KB

How to execute the Adobe Sign Bot - Generate and Refresh Token

If this is the first time running this task, we will need to set some parameters

Open the Server Management Console and locate the imported task

Edit the imported task by right clicking on the task and selecting edit



Tasks

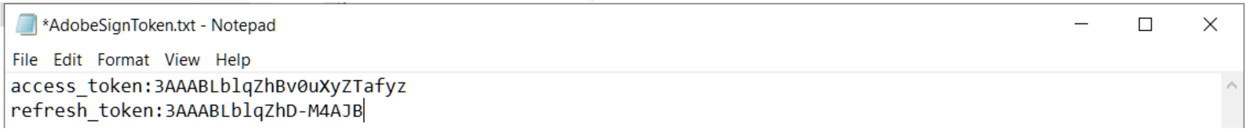
Automate Tasks are automation objects that execute an Automate task on an Automate Task Agent. A task is a collection of steps that perform an automated sequence of actions.

From step 2 edit the next variables according to you desired output:

- var_GetURL: Set the URL of your environment. Adobe Sign uses multiples Endpoints. You can get your URL Endpoint by logging into Adobe Sign and copying the main address domain from your browser address bar. Example: https://secure.na2.echosign.com.
- var_AuthCode: Set the Authorization code. This value is provided when Adobe Sign prompts user for consent. See the user guide for details. This is used just when first running in case a token and the refresh token do not exist.
- var_RedirectUri: The URL for redirection. By default, it is set to **https://localhost**. We don't really need this value for the bot operation, but it is a requisite.
- var_ClientId: This **ID** is unique to your app and is used for initiating OAuth. It is call *Application Integration Key* in Adobe Sign.
- var_ClientSecret: This will be the **Client Secret** generated within the Adobe Sign App that you created.
- var_TokenFileOutput: This will be the path and filename in which you want the bot to create the txt file with the token and token refresh string generated. Default:

  C:\Automate\AdobeSignToken.txt

Here is an example of the content of the text file generated:

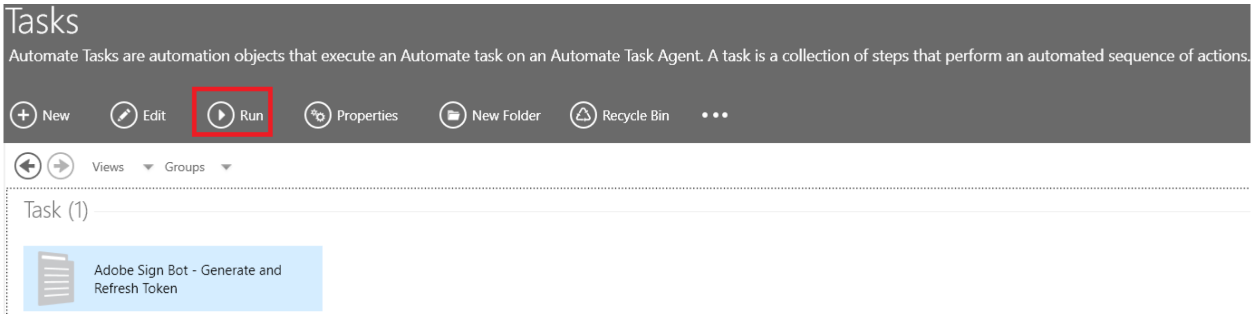


```
access_token:3AAABLblqZhBv0uXyZTafyz
refresh_token:3AAABLblqZhD-M4AJB
```

IMPORTANT*:** If the ***Authorization Code is changed, you need to delete this file before first Bot Execution.

Save and close the task

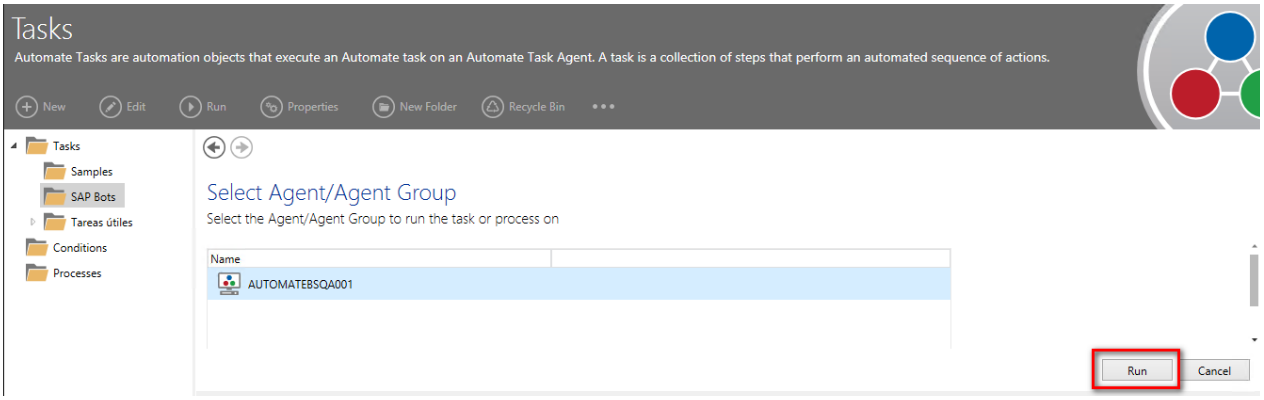
Select the task and click on Run



Tasks

Automate Tasks are automation objects that execute an Automate task on an Automate Task Agent. A task is a collection of steps that perform an automated sequence of actions.

Select your Agent and click on Run again



Tasks

Automate Tasks are automation objects that execute an Automate task on an Automate Task Agent. A task is a collection of steps that perform an automated sequence of actions.

Select Agent/Agent Group

Select the Agent/Agent Group to run the task or process on

Name	
AUTOMATEBSQA001	

Appendix A - Troubleshooting

- Logs: Each iteration of our bot creates a log file for troubleshooting. You can locate the log file under C:\Automate\Tasks\<<TASK NAME>>. By Default, the task name is *Adobe Sign Bot - Generate and Refresh Token.*

www.helpsystems.com

About HelpSystems
Organizations around the world rely on HelpSystems to make IT lives easier and keep business running smoothly. Our software and services monitor and Automate processes, encrypt and secure data, and provide easy access to the information people