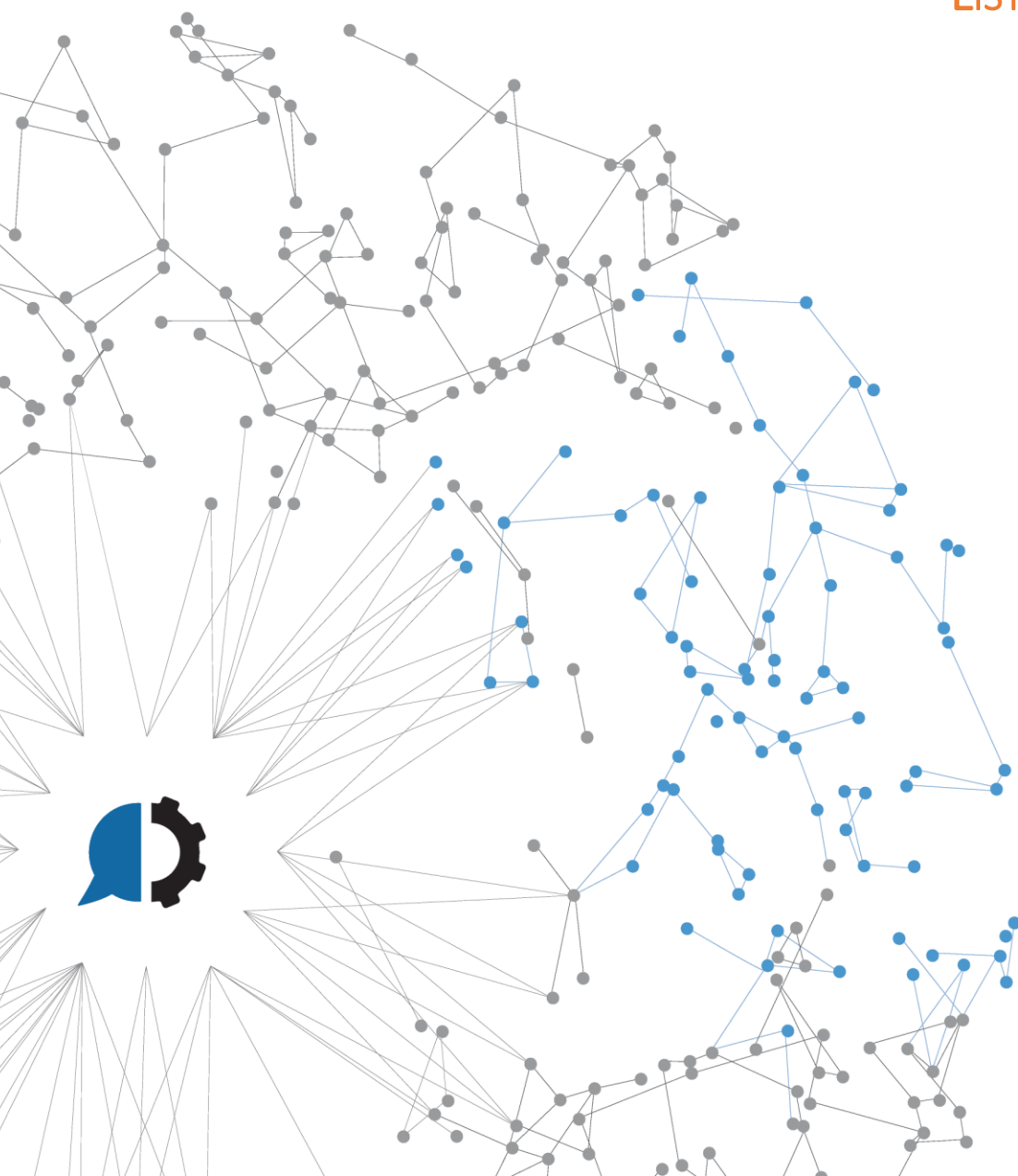




# DOCUSIGN BOT

LIST USERS TO EXCEL





## Content

OVERVIEW .....	3
PREREQUISITES .....	4
HOW TO IMPORT AN AUTOMATE TASK .....	6
HOW TO EXECUTE THE DOCUSIGN BOT - LIST USERS TO EXCEL.....	7
APPENDIX A – DOCUSIGN USERS VIEW AND EXCEL RESULT EXAMPLE.....	9
APPENDIX B - TROUBLESHOOTING .....	10



## OVERVIEW

This how to guide introduces the *DocuSign Bot - List Users to Excel*. 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. With the help of Automate we can take over 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 ***DocuSign Bot - List Users to Excel*** is based in the DocuSign eSignature REST API. It lists the users associated with a particular account that manages them in DocuSign by executing the ***“/accounts/ {accountId} /users”*** HTTP GET method and creates an Excel file with the resulting list of the executions in the task folder. This bot uses the API Token authentication method to avoid using username and password. The token is associated with the account used to scope authorization, so you might need a bot for each account. Check carefully the [PREREQUISITES](#) and the [HOW TO EXECUTE THE DOCUSIGN BOT - LIST USERS TO EXCEL](#) sections to get detailed information.

There is also a [DOCUSIGN BOT - GENERATE AND REFRESH TOKEN](#) available that will provide an easy way for first creation and refreshing of the token for this bot to work. By default, the token expires after 8 hours. This bot is based in the OAuth Authorization Code Workflow.



## PREREQUISITES

- **Automate:** The DocuSign Bot uses Automate software in order to work. The minimum 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 DocuSign Bot task
- **DocuSign:** Our minimum requirements are:
  - **Application for integrations created in the DocuSign eSignature:** You will need to create an application within the DocuSign platform to be able to configure the proper scope and requests for the token creation and use it for the connections of other DocuSign Bots. You will find it by going to your ADMIN area > INTEGRATIONS > API and Keys section. You will need to be an administrator (DS Admin), so if you do not have permissions, ask your DocuSign administrator. More information on how to create the app can be found [here](#).

### API & Keys

Developers who would like to integrate DocuSign into their software need to create and configure an integration key (also known as a client id), as well as other app parameters. An integration key is used to make API calls, whether you are calling an API directly or using an SDK.

#### My Account Information

User ID

API Account ID

Account's Base URI

#### My Apps / Integration Keys

[ADD APP / INTEGRATION KEY](#)

App Name ^	Integration Key	Status	
Automate	f8	● Demo	ACTIONS ▼



- **DocuSign Token generated with proper permissions scopes for the bot execution:** Once the application is created and configured you will be able to generate the token for connection. This token must be generated with the proper permission scopes. For this bot to work you will need the following scopes: ***signature and offline\_acces***. The ***signature*** scope is required for all methods in the eSignature REST API including account management, creating, sending, listing, viewing envelopes, templates, and other objects. The ***offline\_acces*** is for being able to execute the requests in an offline fashion (the way the bot works). There are various OAuth Workflows that you can choose for the Token generation, however, remember that we also provide a ***DocuSign Bot - Generate and Refresh Token*** that's based in the ***DocuSign OAuth Authorization Code Workflow*** that you can use for this matter. It also has a user guide that explains this workflow.
- **DocuSign API Account ID:** This ID is the one associated with your DocuSign account (Platform). You can get it by going to the ADMIN area > INTEGRATIONS > API and Keys section:

**API & Keys**

Developers who would like to integrate DocuSign into their software need to create and configure an integration key (also known as a client id), as well as other app parameters. An integration key is used to make API calls, whether you are calling an API directly or using an SDK.

---

**My Account Information**

User ID

**API Account ID**

**Account's Base URI**

- **DocuSign eSignature API Version:** This bot was created using the DocuSign eSignature API v2.1.
- **Automate constants to be created to set the DocuSign connection values:**
  - **const\_DocuSignURL:** The DocuSign URL for your platform:
    - For the Demo Platform: <https://demo.docusign.net>
    - For the Production Platform: <https://{server}.docusign.net/>

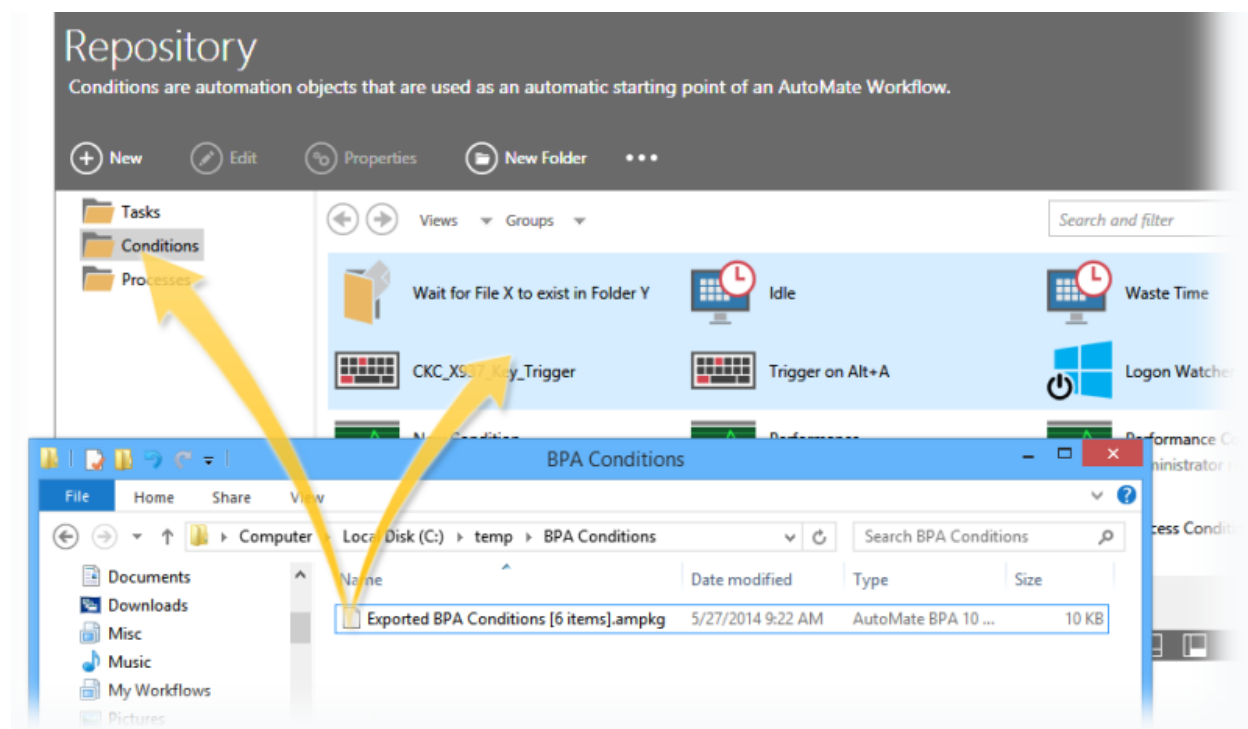


## HOW TO IMPORT AN AUTOMATE TASK

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.



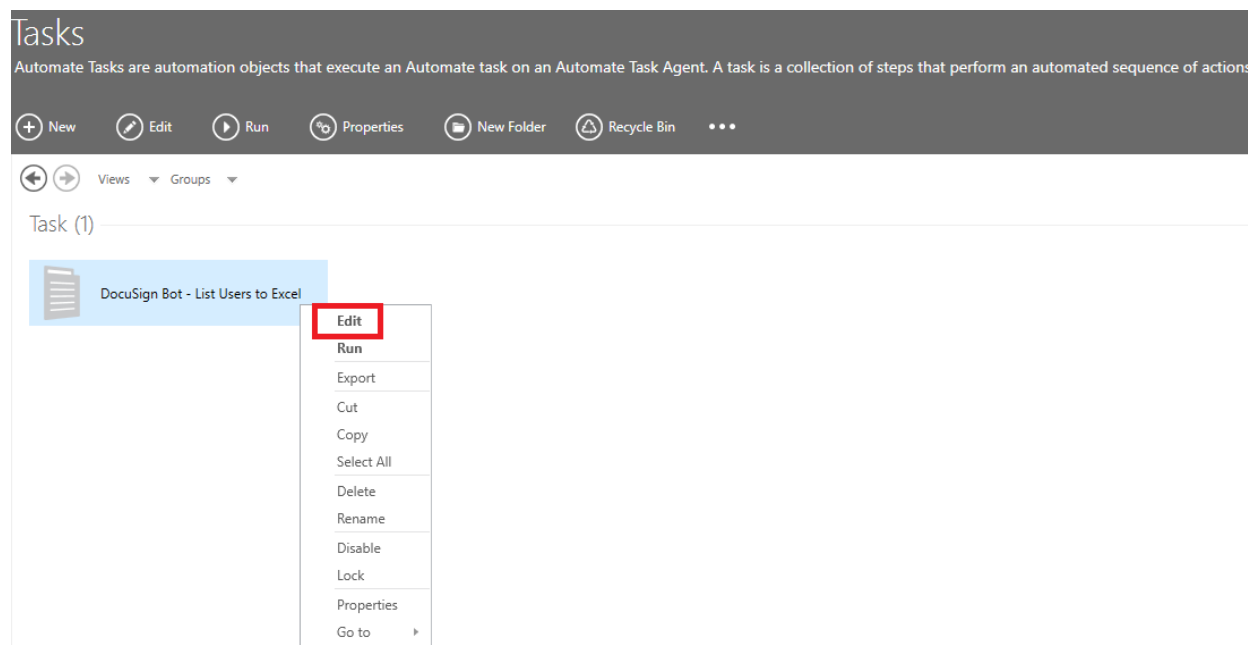


## HOW TO EXECUTE THE DOCUSIGN BOT - LIST USERS TO EXCEL

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](#)



From [step 2](#), edit the next variables according to your desired output:

- **var\_AccountID**: Set the Account ID to which the users are associated. This will be the ID of the user account that manages those users.
- **var\_UserMembershipStatus**: Set the user's membership status. Possible values: ActivationRequired, ActivationSent, Active, Closed and Disabled
- **var\_DocuSignToken**: Set your token for connection (String) or the path of the file with the token information. Remember that the token will be associated to the account we connect with. DocuSign Token to execute the connection/request with. It can be set to the token value itself or the path in which the .txt file with the token inside is located.

Example: **C:\Automate\DocuSignToken.txt**

In the case of using a .txt file, the token must have the next format/structure:

***access\_token:TokenString***



Example of the file:

```
*DocuSignToken.txt - Notepad
File Edit Format View Help
access_token:eyJ0eXAiOiJNVCIzImFsZyI6I1JT
refresh_token:eyJ0eXAiOiJNVCIzImFsZyI6I1JT
```

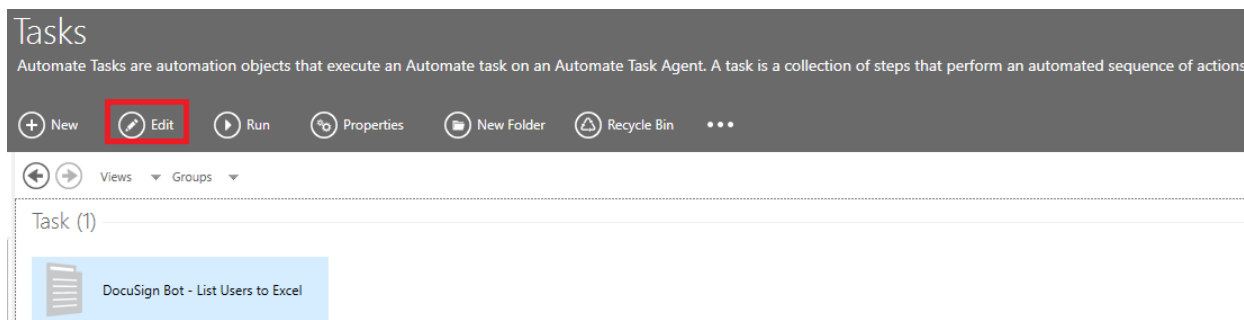
Note that there is a refresh\_token value, this is managed by DocuSign Bot - Generate and Refresh Token

The output file will be automatically generated in the task folder with the next fields:

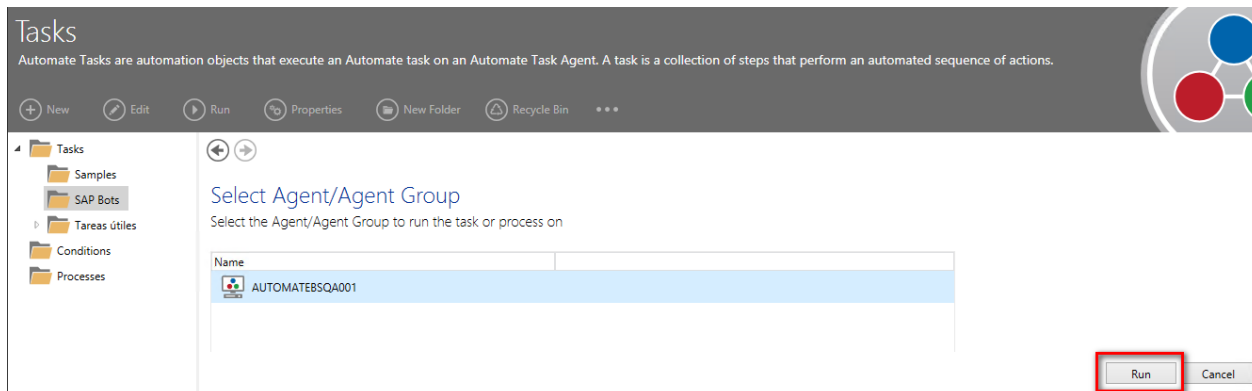
**userId, userName, firstName, lastName, email, jobTitle, company, userType, isAdmin, userStatus, uri, createdDateTime, userAddedToAccountDateTime, permissionProfileId, permissionProfileName and Account ID.**

Save and close the task

Select the task and click on [Run](#)



Select your [Agent](#) and click on [Run](#) again







## APPENDIX A – DOCUSIGN USERS VIEW AND EXCEL RESULT EXAMPLE

### DocuSign Users view:

**Users** Name ▾  FILTERS

Add new users and find, edit, close, or reactivate existing users.

ADD USER DOWNLOAD USERS BULK ACTIONS ▾ 1 - 7 of 7 users First < > Last

Filtered by: Permission Profile (DS Viewer), Status (Closed) | [Reset](#)

Name	Email	Added	Status	Permission Profile	Groups	
Automate 1	AutomateUserOne@testng.com	23/06/2020   16:40:01	● Closed	DS Viewer	Everyone	<span>ACTIONS ▾</span>
Automate 1	AutomateUserOne@testng.com	23/06/2020   16:44:55	● Closed	DS Viewer	Everyone	<span>ACTIONS ▾</span>
Automate 2	AutomateUserTwo@testng.com	23/06/2020   16:44:58	● Closed	DS Viewer	Everyone	<span>ACTIONS ▾</span>

### Excel file with results:

	A	B	C	D	E	F	G	H	I	J	K
1	userid	userName	firstName	lastName	email	jobTitle	company	userType	isAdmin	userStatus	uri
2	ad395965-677c-4985-bfaf-9760e2cd1a37	Automate 3	AutomateUser	Three	AutomateUserThree@testng.com		Personal	CompanyUser	False	Closed	/users/ad395965-677c-4985-bfaf-9760e2cd1a37
3	0b19e20e-bc5d-4d5e-b7f7-f73bf04735d	Automate 1	AutomateUser	One	AutomateUserOne@testng.com		Personal	CompanyUser	False	ActivationSent	/users/0b19e20e-bc5d-4d5e-b7f7-f73bf04735d
4	90f7c3eb-67aa-4979-af0a-67e171d0b7e7	Automate 1	AutomateUser	One	AutomateUserOne@testng.com		Personal	CompanyUser	False	Closed	/users/90f7c3eb-67aa-4979-af0a-67e171d0b7e7

L	M	N	O	P
createdDateTime	userAddedToAccountDateTime	permissionProfileId	permissionProfileName	Account ID
2020-06-23T19:45:01.7430000Z	2020-06-23T19:45:01.7430000Z	10482819	DocuSign Viewer	c90d482c-2a86-41a7-8571-405444ea7057
2020-06-23T19:44:55.3230000Z	2020-06-23T19:44:55.3230000Z	10482819	DocuSign Viewer	c90d482c-2a86-41a7-8571-405444ea7057
2020-06-23T19:40:01.9200000Z	2020-06-23T19:40:01.9200000Z	10482819	DocuSign Viewer	c90d482c-2a86-41a7-8571-405444ea7057

### DocuSignUsers\_25062020\_095537.xlsx

File naming pattern:

DocuSignUsers\_

Base Name

25062020

Represents the date of execution (ddMMyyyy)

095537

Represents the time of execution (Hmss)

**Note:** The output file will be created in the task folder: C:\Automate\Tasks\<<TASK NAME>>.



## APPENDIX B - 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 *DocuSign Bot - List Users to Excel*.



### 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