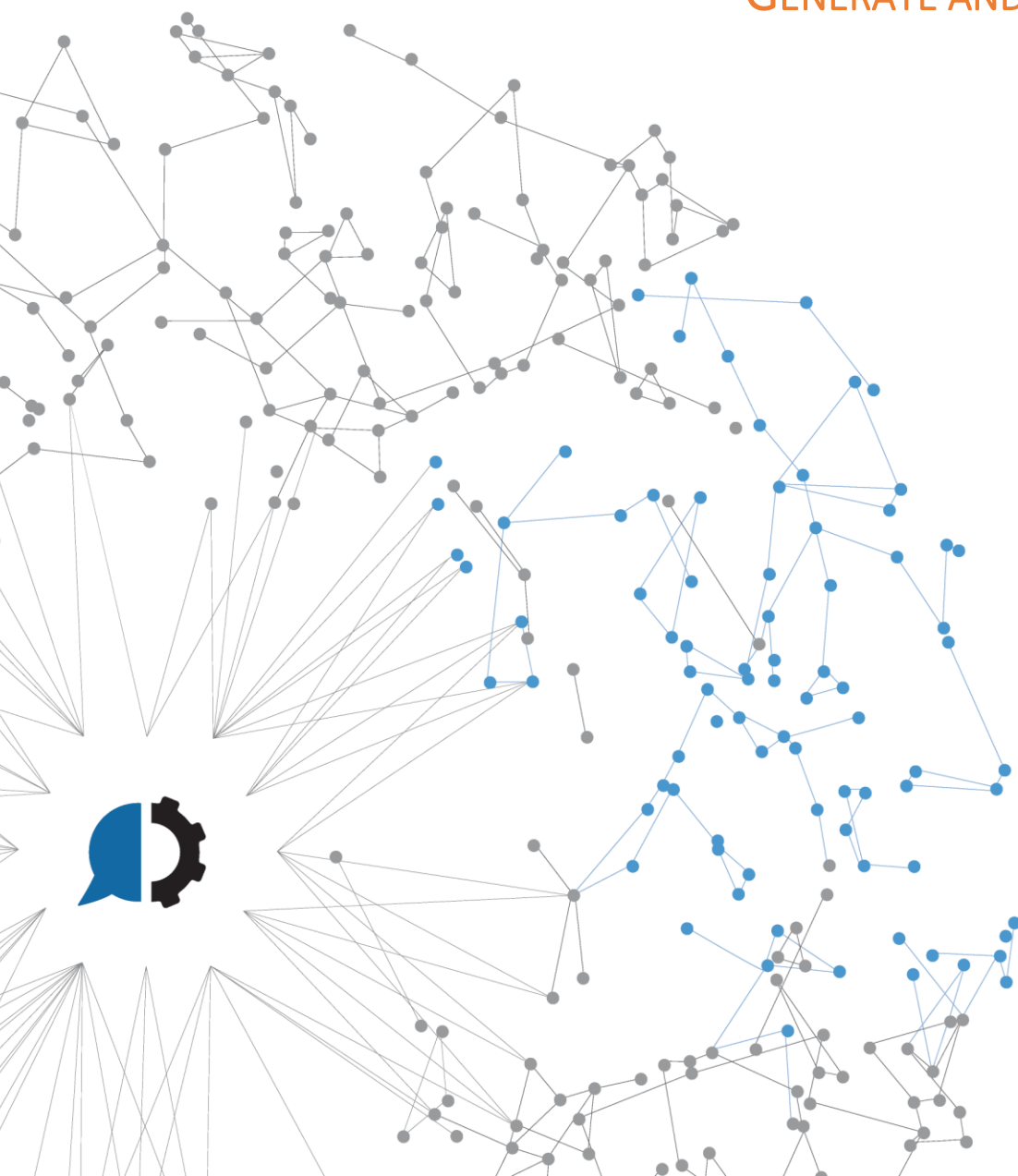




MICROSOFT BOT

GENERATE AND REFRESH TOKEN





Content

OVERVIEW	3
PREREQUISITES	4
HOW TO IMPORT AN AUTOMATE TASK	7
HOW TO EXECUTE THE MICROSOFT BOT - GENERATE AND REFRESH TOKEN	8



OVERVIEW

This how to guide introduces the *Microsoft 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. With the help of Automate we can take over manual steps which greatly reduces the 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.

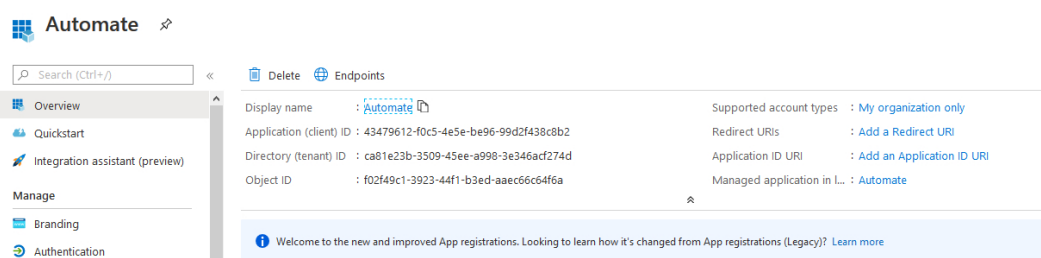
The ***Microsoft Bot - Generate and Refresh Token*** is based in the Microsoft identity platform and the Microsoft Graph API. It generates an **OAuth2** access token requesting it to the <https://login.microsoftonline.com/common/oauth2/v2.0/token>. It uses a “grant_type” parameter with “**authorization_code**” (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. Microsoft does not yet provide a configurable refresh token time, so 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 Microsoft bot work with it.

Check carefully the [PREREQUISITES](#) and the [HOW TO EXECUTE THE MICROSOFT BOT - GENERATE AND REFRESH TOKEN](#) sections to get detailed information.



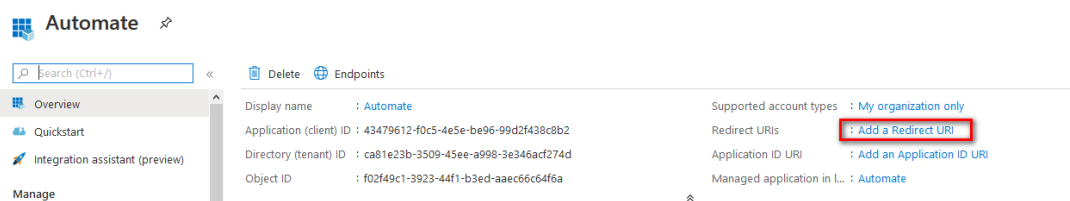
PREREQUISITES

- **Automate:** The Microsoft 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 Microsoft Bot task
- **Microsoft Graph:** Our minimal requirements are:
 - **Registered application with the Microsoft identity platform:** You will need to register an application with the Microsoft identity platform to be able to configure the proper scope and request of the token. You can find more information on how to register at [Microsoft](#).



- **Configure your application to get access to the Microsoft Graph API without a user:** Once you register your applications you will need to configure it to get access without a user. This way you will be able to get a token. In this step you will give permission to the Microsoft Graph API that is used by this bot. You will find the step-by-step on how to do this [here](#).

Some of the configurations you will need are the redirect URL, the supported account types as shown here:





Automate | Authentication

Search (Ctrl+/) Save Discard Got feedback?

- Overview
- Quickstart
- Integration assistant (preview)
- Manage
 - Branding

Platform configurations

Depending on the platform or device this application is targeting, additional configuration may be required such as redirect URIs, specific authentication settings, or fields specific to the platform.

+ Add a platform

Configure platforms

Web applications



Web

Build, host, and deploy a web server application. .NET, Java, Python

Single-page application

Configure browser client applications and progressive web applications. Javascript.

Mobile and desktop applications



iOS / macOS

Objective-C, Swift, Xamarin



Android

Java, Kotlin, Xamarin



Mobile and desktop applications

Windows, UWP, Console, IoT & Limited-entry Devices, Classic iOS + Android

- Overview
- Quickstart
- Integration assistant (preview)
- Manage
 - Branding
 - Authentication
 - Certificates & secrets
 - Token configuration
 - API permissions
 - Expose an API
 - Owners
 - Roles and administrators (Preview)
 - Manifest
- Support + Troubleshooting
 - Troubleshooting
 - New support request

Platform configurations

Depending on the platform or device this application is targeting, additional configuration may be required such as redirect URIs, specific authentication settings, or fields specific to the platform.

+ Add a platform

Web

Quickstart

Docs



Redirect URIs

The URIs we will accept as destinations when returning authentication responses (tokens) after successfully authenticating users. Also referred to as reply URLs. [Learn more about Redirect URIs and their restrictions](#)

http://localhost

Add URI

Logout URL

This is where we send a request to have the application clear the user's session data. This is required for single sign-out to work correctly.

e.g. <https://myapp.com/logout>

Implicit grant

Allows an application to request a token directly from the authorization endpoint. Checking Access tokens and ID tokens is recommended only if the application has a single-page architecture (SPA), has no back-end components, does not use the latest version of MSAL.js with auth code flow, or it invokes a web API via JavaScript. ID Token is needed for ASP.NET Core Web Apps. [Learn more about the implicit grant flow](#)

To enable the implicit grant flow, select the tokens you would like to be issued by the authorization endpoint:

- ☐ Access tokens
- ☐ ID tokens

Supported account types

Who can use this application or access this API?

- ☐ Accounts in this organizational directory only (HelpSystems only - Single tenant)
- ☒ Accounts in any organizational directory (Any Azure AD directory - Multitenant)



It is important to give the Microsoft Graph the proper scope for this bot to work. In this case it will be **Calendars.Read**. As shown in the next picture:

OutlookCalendarforAutomate | API permissions

Search (Ctrl+/) Refresh

Overview
Quickstart
Integration assistant (preview)

Manage

Branding
Authentication
Certificates & secrets
Token configuration
API permissions

Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. [Learn more about permissions and consent](#)

+ Add a permission Grant admin consent for HelpSystems

API / Permissions name	Type	Description	Admin consent req...	Status
Microsoft Graph (2)				...
Calendars.Read	Delegated	Read user calendars	-	...
User.Read	Delegated	Sign in and read user profile	-	...

Generate an application client Secret with the desired expiration date:

Certificates & secrets

Token configuration
API permissions
Expose an API
Owners
Roles and administrators (Previe...
Manifest

Client secrets

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

+ New client secret

Description	Expires	Value
Automate	5/19/2021	_nB*****



Here you can check the overview result of the configured app:

Search (Ctrl+J)	Delete	Endpoints
Overview	Display name : OutlookCalendarforAutomate	Supported account types : Multiple organizations
Quickstart	Application (client) ID : a993a231	Redirect URIs : 1 web, 0 spa, 0 public client
Integration assistant (preview)	Directory (tenant) ID : ca81e23b	Application ID URI : Add an Application ID URI
Manage	Object ID : ca21a018-	Managed application in ... : OutlookCalendarforAutomate

Authorization Code for first execution: Once the application is registered and configured you will be able to get the Authorization Code needed for first time Bot Execution. For this you will need to open the next URL in your browser (change the **##CLIENTID##** with yours) and you will receive the Authorization code in your browser address bar. Note that in the URL we already specify the scope needed. In this example: *Calendars.Read*. You need to log into your Outlook account.

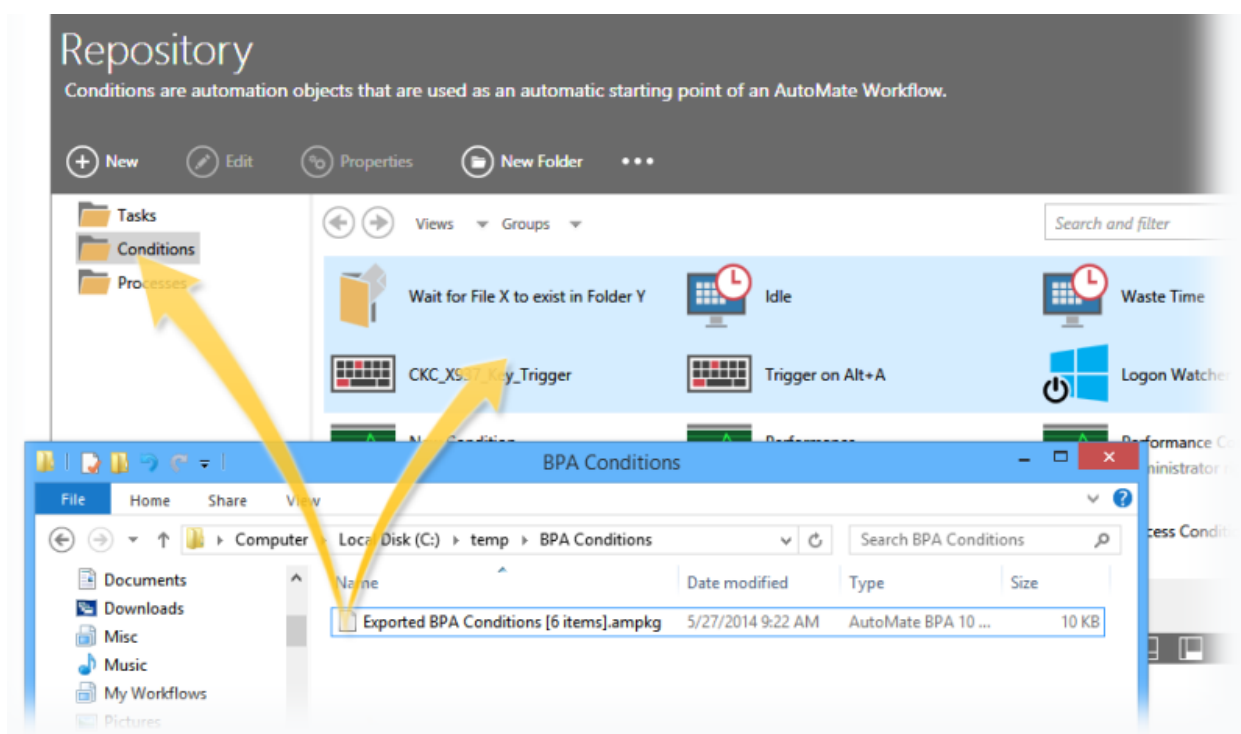
https://login.microsoftonline.com/common/oauth2/v2.0/authorize?client_id=##CLIENTID##&response_type=code&scope=Calendars.Read

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.

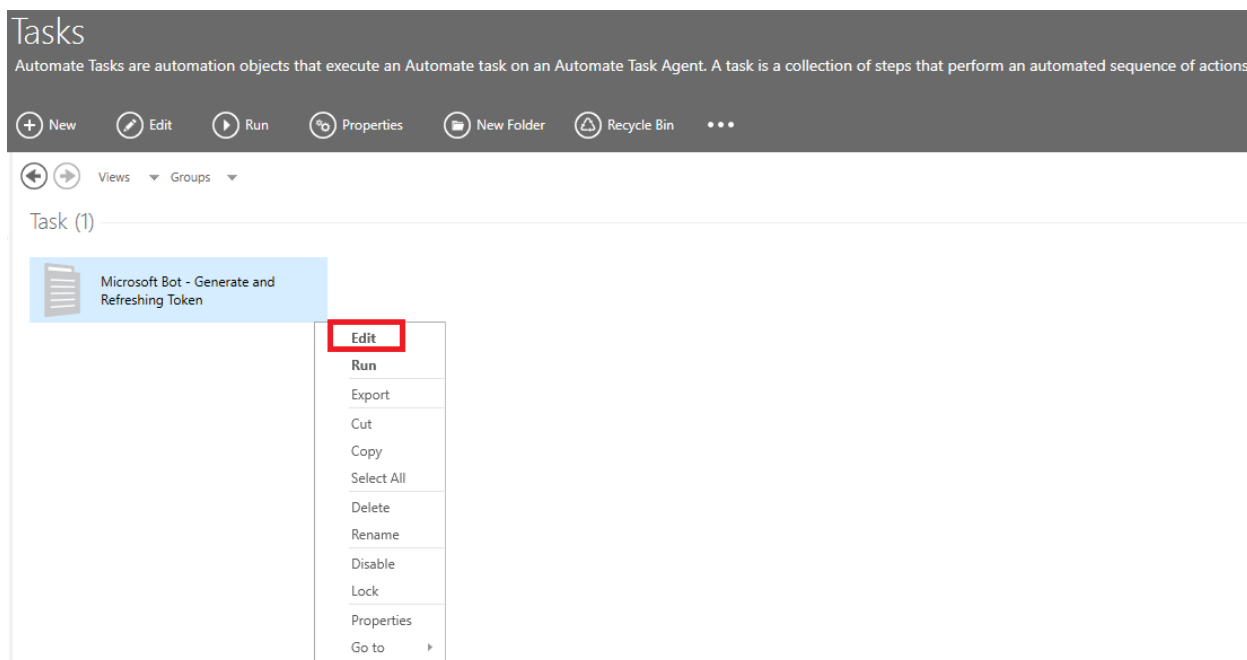


HOW TO EXECUTE THE MICROSOFT BOT - GENERATE AND REFRESH TOKEN

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](#)



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

- **var_GetTokenURL**: The URL used to request a Microsoft Token generation. It is set to the default URL provided by Microsoft:

<https://login.microsoftonline.com/common/oauth2/v2.0/token>

- **var_AuthCode**: This is the Authorization Code received during the step-by-step provided by Microsoft (mentioned in the prerequisites). This is used just when first running in case a token and refresh_token do not exist.
- **var_RedirectUri**: The URL for redirection. By default, it is set to <http://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.
- **var_ClientSecret**: This will be the **Client Secret** generated within the Microsoft 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\MicrosoftGraphToken.txt



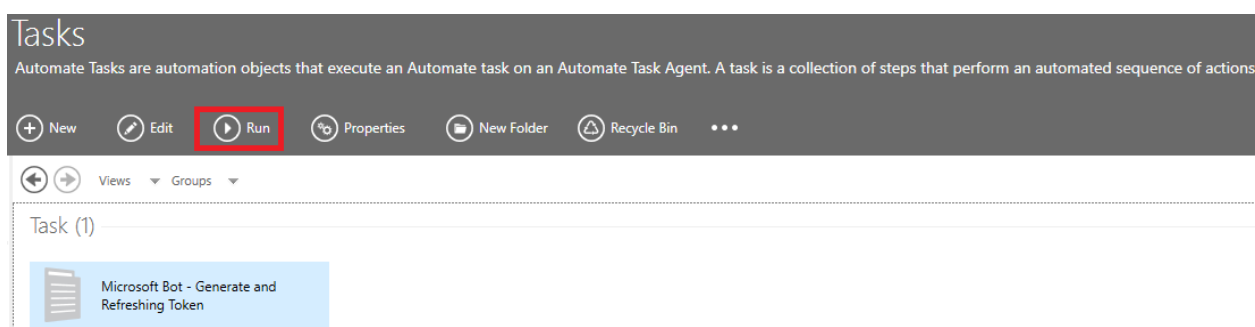
Here is an example of the content of the text file generated:

```
*MicrosoftGraphToken.txt - Notepad
File Edit Format View Help
access_token:eyJ0eXAiOiJKV1QiLCJub25jZSI6I1RCTmF4d3o0NXJxR1Y0cmd3d1Bwbmt
refresh_token:OAQABAAAAAAAm-06b1BE1TpVMi18KPQ41K3Uz21ceZ28yXAzAEFSdFyI7s|
```

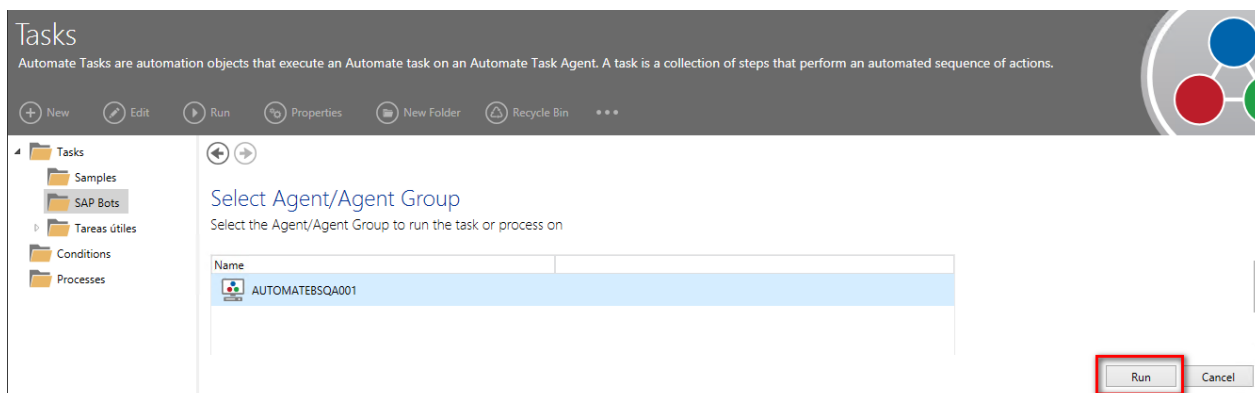
IMPORTANT: If the **Authorization Code** is changed, you need to delete this file before first Bot Execution.

Save and close the task

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



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





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 *Microsoft Bot - Generate and Refreshing Token*.



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