

09.11.2022

# Read Emails and Attachment from Office 365 using IDMC (CAI)

Rahul Minda

*Senior Consultant*  
*Informatica Professional Service*

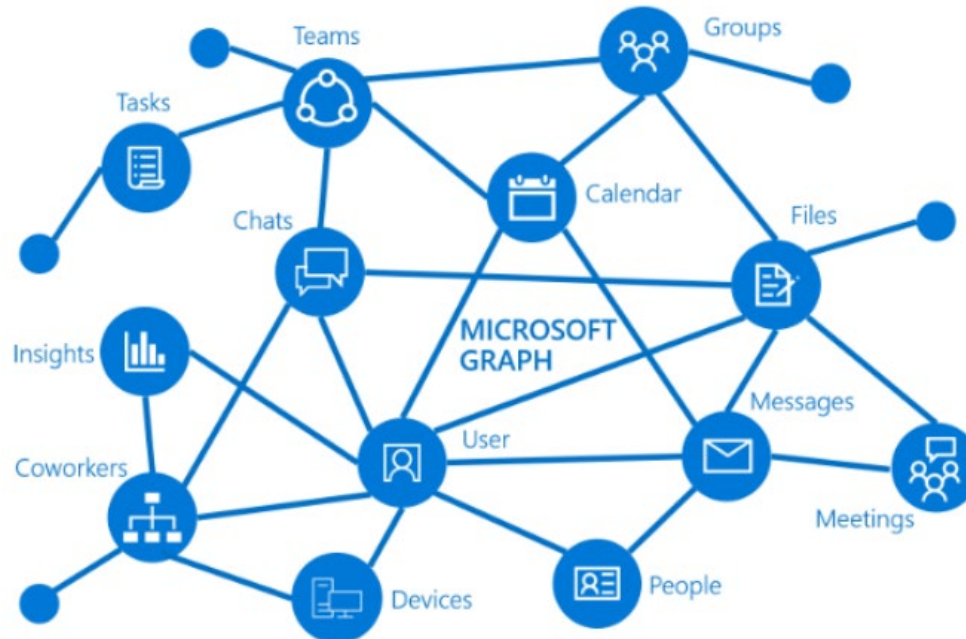
# Agenda

- Microsoft Graph API Overview
- Microsoft Graph API Integration with IDMC
- Prerequisites: Azure Portal Setup in office 365 sandbox
- Prerequisites: Token Generation
- Connection setup within CAI
- Defining Input Parameters for the custom API
- Demo
- Q&A

# Microsoft Graph API Overview

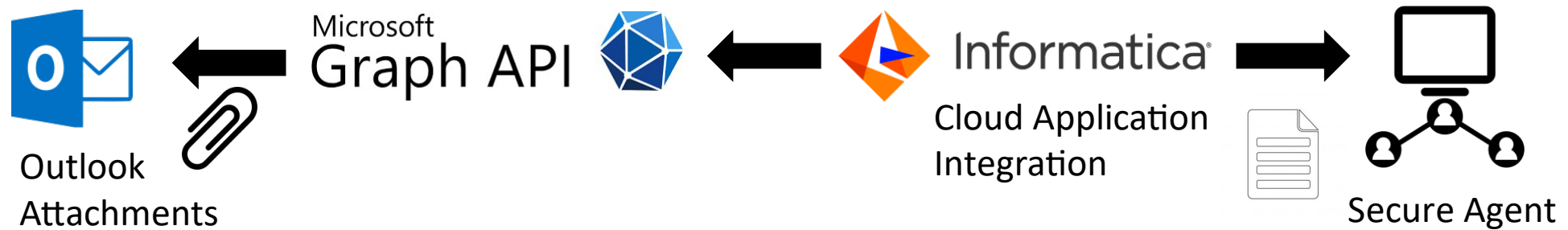
The Microsoft Graph API offers a single endpoint, <https://graph.microsoft.com>, to provide access to rich, people-centric data and insights in the Microsoft cloud, including Microsoft 365, Windows, and Enterprise Mobility + Security. One can use REST APIs or SDKs to access the endpoint and build apps that support Microsoft 365 scenarios, spanning across productivity, collaboration, education, people, and workplace intelligence, and more.

Microsoft Graph exposes REST APIs and client libraries to access data on the following Microsoft cloud services:



# Microsoft Graph API Integration with IDMC

- Cloud Application Integration (CAI) service offers a single, trusted solution to support any integration pattern, data set, user-type, or endpoint to automate business processes, expedite transactions and enable real-time analytics.
- The utility developed on CAI will handle all the complexity of traversing multiple Microsoft Graph APIs in the backend and presents the user with some basic Input requests with which an attachment from an email can be fetched and copied over to the Secure Agent server directory.



# Prerequisites: Azure Portal Setup in office 365 sandbox

Microsoft Graph API requires certain permissions to be given to it on Azure portal and requires some additional steps to ensure safety and adherences to security. The article below covers everything about authorization and security in details.

<https://learn.microsoft.com/en-us/graph/auth/auth-concepts?view=graph-rest-1.0>

**Registration to Azure Portal:** Azure portal registration is an important step as it integrates your application with Azure. Link: <https://portal.azure.com/>

**API Permissions:** Once registered it is critical to provide Microsoft Graph API some permissions. Read Permissions are a must and write permissions can be skipped if there is no plan to customize the API further for specific needs.

Refresh

Got feedback?

Add a permission

Grant admin consent for 83trf2

API / Permissions name	Type	Description	Admin consent req...	Status
▼ Microsoft Graph (7)				
APIConnectors.Read.All	Application	Read API connectors for authentication flows	Yes	✔ Granted for 83trf2
APIConnectors.ReadWrite.All	Application	Read and write API connectors for authentication flows	Yes	✔ Granted for 83trf2
Mail.Read	Application	Read mail in all mailboxes	Yes	✔ Granted for 83trf2
Mail.ReadBasic	Application	Read basic mail in all mailboxes	Yes	✔ Granted for 83trf2
Mail.ReadBasic.All	Application	Read basic mail in all mailboxes	Yes	✔ Granted for 83trf2
Mail.ReadWrite	Application	Read and write mail in all mailboxes	Yes	✔ Granted for 83trf2
Mail.Send	Application	Send mail as any user	Yes	✔ Granted for 83trf2

# Prerequisites: Token Generation

Token Generation is required to ensure adherence to security protocols and to ensure no unauthorized access takes place. The utility can generate token if it is provided with the necessary keys as an encrypted value at the connections level.

**Client ID:** It is the application ID that registration portal has assigned the app. Can be checked in following link:

<https://go.microsoft.com/fwlink/?linkid=2083908>

**Tenant ID:** It can be located by searching for tenant properties in Azure and then scrolling down to locate tenant ID.

[https://portal.azure.com/#view/Microsoft\\_AAD\\_IAM/TenantPropertiesBlade](https://portal.azure.com/#view/Microsoft_AAD_IAM/TenantPropertiesBlade)

The screenshot displays the Azure App Registrations portal. At the top, there's a navigation bar with options like 'New registration', 'Endpoints', 'Troubleshooting', 'Refresh', 'Download', 'Preview features', and 'Got feedback?'. Below this, a message states that starting June 30th, 2020, new features for ADAL and Azure AD Graph will no longer be added, and applications will be upgraded to MSAL and Microsoft Graph. The main section shows 'Owned applications' with a search bar and 'Add filters' button. One application is listed: 'Graph Java quick start' with a red icon, a redacted client ID, and a creation date of 9/15/2022. Below the application list, the 'Tenant Properties' blade is open, showing details for the tenant '83trf2'. The properties include Country or region (Singapore), Location (Asia, United States, Europe datacenters), Notification language (English), and Tenant ID (redacted).

# Prerequisites: Automating Token Generation

**Client Secret:** Client Secret key can be created under Certificates & Secrets tab in Azure directory. The 'value' field defined is the actual key which should be used to generate token.

Except for Client Secret all other are static values. Client Secret should be newly created every few months as a best practice to ensure security adherence.

Search

Got feedback?

Branding & properties

Authentication

**Certificates & secrets**

Token configuration

API permissions

Expose an API

App roles

Owners

Roles and administrators

Manifest

Support + Troubleshooting

Troubleshooting

Application registration certificates, secrets and federated credentials can be found in the tabs below.

Certificates (0) **Client secrets (2)** Federated credentials (0)

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 ⓘ	Secret ID
test API	3/15/2023	[REDACTED]	[REDACTED] [Copy] [Delete]
API	3/16/2023	[REDACTED]	[REDACTED] [Copy] [Delete]

# Connection setup within CAI

To use the utility, one needs to import it within Informatica Cloud Application Integration and update the connections within it.

**Update of App Connection:** The App connection will require the values of Client ID, Tenant ID and Client Secret to generate the token and call the Microsoft Graph API as required. Once it is updated alongside repointing of Secure Agent then Publish the connection.

The screenshot shows the 'Definition' tab for a custom connector named 'sc-Microsoft-Graph-API-custom'. The connector is marked as 'Valid'. The 'Name' field is 'sc-Microsoft-Graph-API-custom' and the 'Location' is 'Rahul'. The 'Description' field is empty. The 'Agent Only' checkbox is unchecked. Below the form is a table for 'Connection Properties'.

Name	Description	Test With	Type	Required	Encrypt	
tenantid	tenantid	.....	string	✓	✓	✕
client_id	client_id	.....	string	✓	✓	✕
client_secret	client_secret	.....	string	✓	✓	✕

The screenshot shows the 'Properties' tab for an application connection named 'AppConn-Microsoft-Graph-API-custom'. The connection is marked as 'Valid'. The 'Description' field is empty. The 'Type' is 'Rahul > sc-Microsoft-Graph-API-custom'. The 'Run On' is 'Rahul\_INWPF2H7X55-AAD'. The 'Connection Test' and 'OData-Enabled' status are both 'Not Supported'. Below the form is a table for 'Connection Properties'.

Name	Value	Description
tenantid:*	.....	tenantid
client_id:*	.....	client_id
client_secret:*	.....	client_secret



# Connection setup within CAI

**Update of File Connection:** Two files are generated as part of the process; one being the data file and other being the Delta Key file. One needs to repoint the Secure Agent and update the Target Directory for file generation under Event Target tab.

- Delta key file stores the key using which Delta Detection is done on emails and only the latest emails are processed. This ensures processing of only latest files during every run.
- Data file from email attachment is recreated on Secure Agent server if the previous file is not present and is appended to existing file if a file is already present on Secure Agent server. This solution is used to handle multiple attachment files within one single email.

The screenshot shows the 'AppConn-Microsoft-Graph-API-Custom-File-Generator' window with the 'Properties' tab selected. The 'Connection Details' section includes the following fields:

- Name: AppConn-Microsoft-Graph-API-Custom-Fi (Unpublish connection to edit name)
- Location: Rahul (with a 'Browse...' button)
- Description: (empty text area)
- Type: File (dropdown menu)
- Run On: Rahul\_INWPF2H7X55-AAD (dropdown menu)
- Connection Test: (button)
- OData-Enabled: Not Supported

The screenshot shows the 'AppConn-Microsoft-Graph-API-Custom-File-Generator' window with the 'Event Targets' tab selected. The 'FileWriter : FileWriter' section includes the following fields:

- Event: (dropdown menu)
- Name: FileWriter (text input)
- Description: Use this event target to write output to a file. (text area)
- Properties: (table with 3 columns: Name, Value, Description)
- File Location Settings: Directory: C:\File\_receiver (text input) with description 'Directory that should be used to store files.'
- File Exists Settings: (empty section)

# Defining Input Parameters for the custom API

**userID:** Should be the email/user id of the receiver.

**SearchFolder:** Pass a search string to locate the email in a particular folder

**SearchEmailSubject:** Pass the email subject as a search string with which the folder is searched for

**SearchEmailID:** Pass the sender email ID which would contain the data file

**outputFileName:** To provide the file name which would be generated with data from attachments in email

**DeltaFilePath:** To provide the file path which you would have specified in the Delta file connection

**DeltaFileName:** To pass the file name which would have the delta key stored for next run.

The screenshot displays the Informatica Integration Hub interface for a custom API process named "Microsoft-Graph-API-Custom-Process-Attachments". The "Design" tab is active, showing a table of input and output fields. The "Input Fields" section is expanded, listing fields such as userID, SearchFolder, SearchEmailSubject, SearchEmailID, outputFileName, DeltaFilePath, and DeltaFileName. Each field is marked as "Required".

Name	Type	Description	Required
userID	Text		✓
SearchFolder	Text		✓
SearchEmailSubject	Text		✓
SearchEmailID	Text		✓
outputFileName	Text		✓
DeltaFilePath	Text		✓
DeltaFileName	Text		✓

Below the table, the "Process Input" section shows the JSON input data:

```
{
  "userID": "rminda@83trf2.onmicrosoft.com",
  "SearchFolder": "Inbox",
  "SearchEmailSubject": "API TESTING",
  "SearchEmailID": "rminda@informatica.com",
  "outputFileName": "test_graph_api.csv",
  "DeltaFilePath": "C:/File_receiver",
  "DeltaFileName": "Microsoft_Graph_API_Delta_Value.txt"
}
```

# DEMO



# Q&A

# Thank you