

Data on Demand Addendum Management Pack for Windows Computers

Introduction

The MP named [Windows.Computer.DataOnDemand.Addendum](#) provides different kinds of Agent Tasks for direct use in the SCOM console or in a Dashboard Tool.

Tasks

- Get-ComputerDescription
 - Retrieves the computer's local description and the one from the Active Directory computer object.
- Get-ComputerLastChanges
 - Displays the computers' last changes; last installed software, interactive logged on user, last boot, etc.
- Get-RemoteSOAPServiceInfo
 - Queries a remote web service via SOAP / XML displays its data.
 - Before RESTful became standard, web services could be consumed via SOAP.
- Get-SCOMNotificationConfig
 - Displays the SCOM's notification configuration incl. subscription, subscribers, channel, etc. → Tasks works only if selecting a Management Server!
- Get-MSGraphData
 - Queries Microsoft's Graph API by extending given OData functions
- Get-MSGraphBetaData
 - As Get-MSGraph Data, but targeting the beta version of the API.
- Get-MSO365MgmtData
 - Queries Microsoft's O365 Management API to get retrieve Service Health, Incidents and other information

Change History

Date	Build No.	Changes
2020-01-28	1.0.0.93	Initial upload to GitHub
2020-07-22	1.0.0.166	Added tasks for Graph API
2020-08-17	1.0.0.181	Added task for O365 Management API
2020-10-18	1.0.0.193	Support for pagination in Graph API Adding CPU & RAM info to Get-ComputerDescription

Contents

Introduction.....	1
Tasks	1
Change History	2
Overview.....	2
Tasks	3
Get-ComputerDescription	3
Get- ComputerLastChanges.....	4
Get- SCOMNotificationConfig.....	5
Get-RemoteSOAPServiceInfo	6
Get- MSGraphData / Get-MSGraphBetaData.....	7
Get-MSO365MgmtData.....	8
Preparation.....	9
Retrieve data from MS Graph API	9
Retrieve data from MS O365 Management API.....	13

Overview

This Management Pack does not require a new MP for storing overrides.

Tasks

Get-ComputerDescription

Description:

Retrieves the computer's local description and the one from the Active Directory computer object.

Possible overrides:

- Output format: CSV, JSON, Text. Default is CSV
- Script timeout in seconds. Default is 180.

Screenshot:

The screenshot shows a task execution window titled "Task Output". At the top right, there are buttons for "Copy Text" and "Copy HTML". The task is identified as "Get-ComputerDescription (Windows Computer DataOnDemand Addendum)" with a green checkmark icon. The task status is "Success".

Task Description: Displays the Computer's local description and the one from AD. Note: JSON format is only supported if PowerShell v3 or later is installed on the target server.

Task Output:

```
"AD Computer CreationDate", "AD Computer Description", "Local Computer Description"
"10/12/2015 03:15:47", "Compliance Database PSA//prod", "compliance database psa//prod"
```

Get- ComputerLastChanges

Description:

Displays the computers' last changes; last installed software, interactive logged on user, last boot, etc.

Possible overrides:

- Output format: CSV, JSON, Text. Default is CSV
- Script timeout in seconds. Default is 180.

Screenshot:

The screenshot shows a 'Task Output' window with a title bar containing 'Copy Text' and 'Copy HTML' buttons. The main content area is divided into two sections: 'Task Description' and 'Task Output'.

Task Description:

- Task:** Get-ComputerLastChanges (Windows Computer DataOnDemand Addendum)
- Status:** Success
- Scheduled Time:** 1/27/2020 3:57:18 PM
- Start Time:** 1/27/2020 3:57:18 PM
- Submitted By:** [User Icon]
- Run As:** [User Icon]
- Run Location:**
- Target:**
- Target Type:** Windows Computer
- Category:** Operations

Task Output:

```
"Last Software Install", "Last Interactive Logon", "Last Hotfix Install", "Patch-Boot pending", "Last Boot / UpTime (days)"
"Adobe Acrobat Reader DC / 2020-01-07", " / 2019-
```

Get- SCOMNotificationConfig

Description:

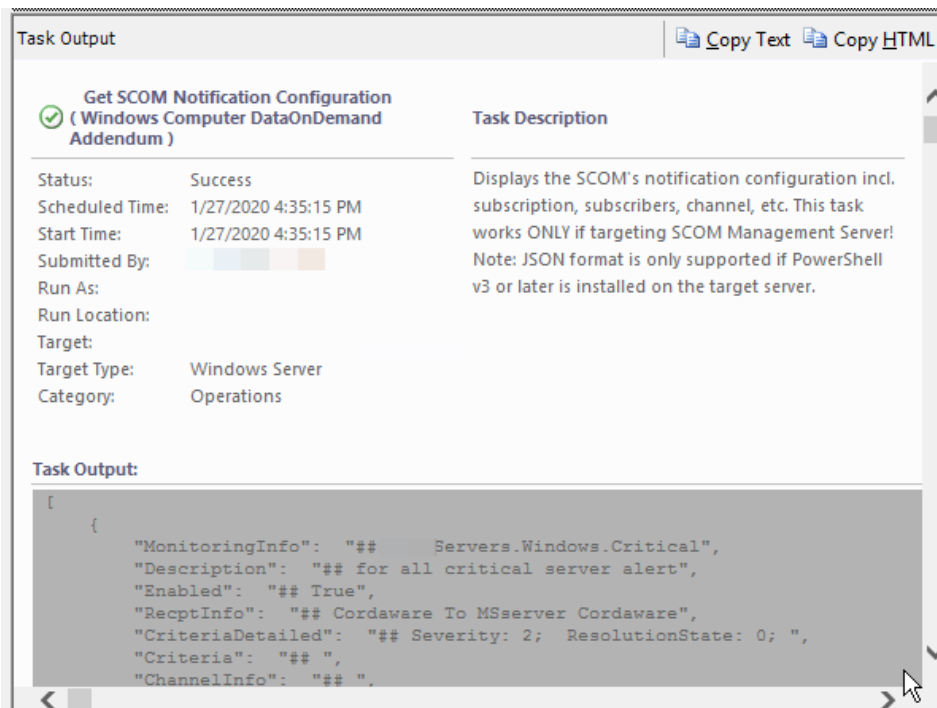
Displays the SCOM's notification configuration incl. subscription, subscribers, channel, etc.

→ Task works only if selecting a SCOM Management Server!

Possible overrides:

Output format	CSV, JSON, Text. Default is CSV
TimeOut	Script timeout in seconds. Default is 180.
HideDisabled	Hides disabled subscriptions [true false] (default = true)
HideChannelInfo	Hides Channel Information [true false] (default = true)
SortedByColumn	Sorts according to the field specified. (E.g. Description, DisplayName, RecptInfo, Criteria, Enabled)
AddVisualizationFlag	Adds a ## (double hashtag) at the beginning of every 2nd entry so that nice formatting can be performed

Screenshot:



Get-RemoteSOAPServiceInfo

Description:

Queries a remote web service via SOAP / XML displays it's data.

Possible overrides:

Output format	JSON, Text, List, default: JSON
TimeoutSeconds	Script timeout, default: 180
WebServiceURL	e.g.: https://webserver.domain.com/apipoint
ContentType	Default: application/soap+xml
XMLNodeName	Name of the single node (e.g. unit or item or ...)
UserName	UserName in clear text, used for basic authentication
PassWord	Password in clear text, used for basic authentication
SortedByXMLNode	Filter a Node with PowerShell syntax. E.g. Element -notLike "*off*"
Headers	Request headers; presently only automatic via UserName and PassWord
AddVisualizationFlag	Adds a ## (double hashtag) at the beginning of every 2nd entry so that nice formatting can be performed. Default is: True

Screenshot:

The screenshot shows a 'Task Output' window with a title bar and buttons for 'Copy Text' and 'Copy HTML'. The main content area is divided into two sections: 'Task Description' and 'Task Output'.

Task Description:

- Get-RemoteSOAPServiceInfo (Windows Computer DataOnDemand Addendum)**
- Status:** Success
- Scheduled Time:** 1/27/2020 4:23:26 PM
- Start Time:** 1/27/2020 4:23:26 PM
- Submitted By:** [User Icon]
- Run As:** [User Icon]
- Run Location:** [User Icon]
- Target:** [User Icon]
- Target Type:** Windows Computer
- Category:** Operations

Task Output:

```
[
  {
    "ChannelName": "## cc_dubai_ftp",
    "ShortLog": "## ShortLog",
    "Counter": "## 0",
    "Direction": "## OUTBOUND",
    "AdapterType": "## File",
    "Party": "## CBOB",
    "ChannelID": "## e93f90e224733f36bec9e8cf31ab46d0",
    "ActivationState": "## STOPPED",
    "Control": "## MANUAL",
    "Service": "## dubai ftp"
  }
]
```

Get- MSGraphData / Get-MSGraphBetaData

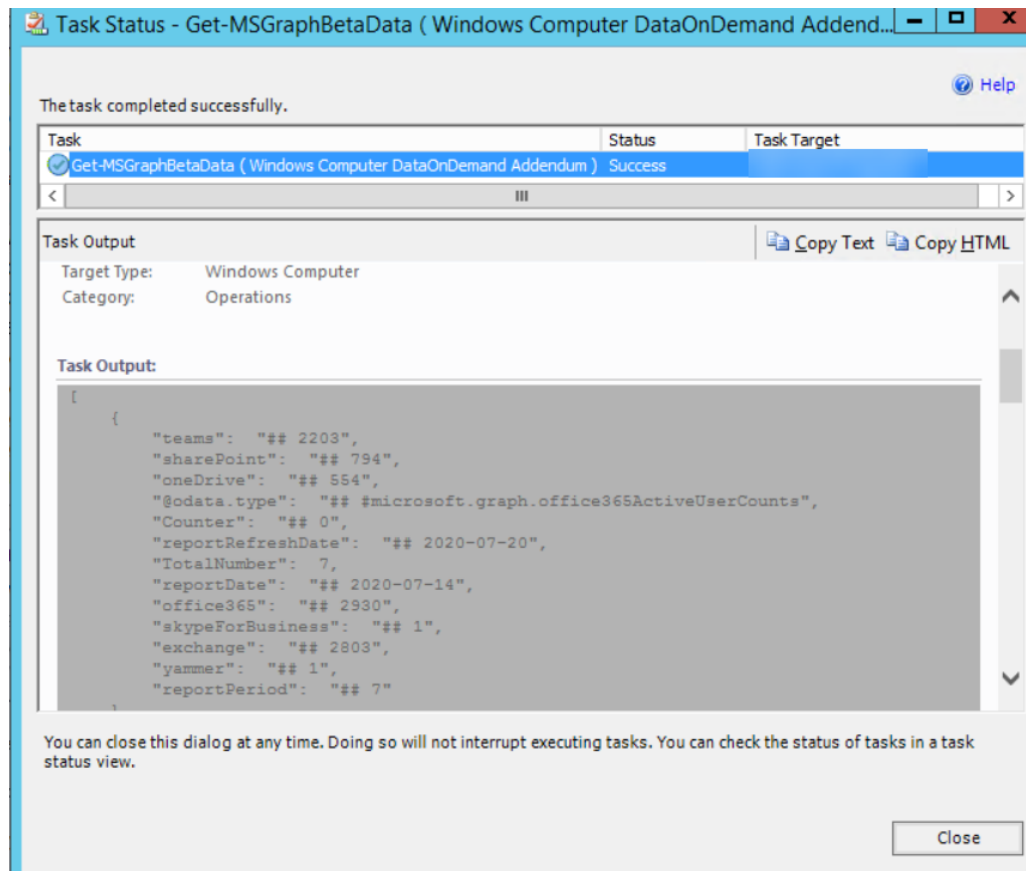
Description:

Queries Microsoft's Graph API by extending given OData functions

Possible overrides:

Output format	JSON, Text, List, default: JSON
TimeoutSeconds	Script timeout, default: 180
WebServiceURL	Used for current Graph API Urls, overridable if MS changes.
ClientSecret	Retrieved from the Azure Portal / App Registration
ClientId	Retrieved from the Azure Portal / App Registration
TenantId	Retrieved from the Azure Portal / App Registration
FilteredBy	Filtering query result by PowerShell syntax
SortedBy	Property of the query that is used to sort the result
SortDescending	Flag to decide if result is sorted from Top to Bottom
DisplayItemNumber	Limits the items to be shown as the result
GraphQry	Query for the Graph API
AddVisualizationFlag	Adds a ## (double hashtag) at the beginning of every 2nd entry so that nice formatting can be performed. Default is: True

Screenshot:



Get-MSO365MgmtData

Description:

Queries Microsoft's O365 Management API

Possible overrides:

Output format	JSON, Text, List, default: JSON
TimeoutSeconds	Script timeout, default: 180
WebServiceURL	Used for current API Urls, overridable if MS changes.
ClientSecret	Retrieved from the Azure Portal / App Registration
ClientId	Retrieved from the Azure Portal / App Registration
TenantId	Retrieved from the Azure Portal / App Registration
FilteredBy	Filtering query result by PowerShell syntax
SortedBy	Property of the query that is used to sort the result
SortDescending	Flag to decide if result is sorted from Top to Bottom
DisplayItemNumber	Limits the items to be shown as the result
GraphQry	Query for the Graph API
AddVisualizationFlag	Adds a ## (double hashtag) at the beginning of every 2nd entry so that nice formatting can be performed. Default is: True

Screenshot:

The task completed successfully.

Task	Status	Task Target
Get-MSO365MgmtData (Windows Computer DataOnDemand Addendum)	Success	

Task Output

```
[
  {
    "Classification": "## Advisory",
    "FeatureDisplayName": "## Management and Provisioning",
    "Messages": "## ",
    "Severity": "## Sev2",
    "Title": "## Any user with end-user spam notification (ESN) policies using s",
    "AffectedUserCount": "## ",
    "MessageType": "## Incident",
    "ImpactDescription": "## Users with an ESN policy scoped by a rule using spe",
    "TenantParams": "## ",
    "WorkloadDisplayName": "## Exchange Online",
    "Feature": "## Provisioning",
    "Workload": "## Exchange",
    "AdditionalDetails": "## ",
    "Status": "## Extended recovery",
    "Id": "## EX213379",
    "UserFunctionalImpact": "## ",
    "AffectedTenantCount": "## 0",
    "EndTime": "## ",
    "AffectedWorkloadNames": "## ",
    "LastUpdatedTime": "## 2020-08-14T20:46:55.052",
    "FilteredNumber": 0,
    "StartTime": "## 2020-05-15T18:00:32Z",
```


Preparation

This Management Pack does not require a new MP for storing overrides.

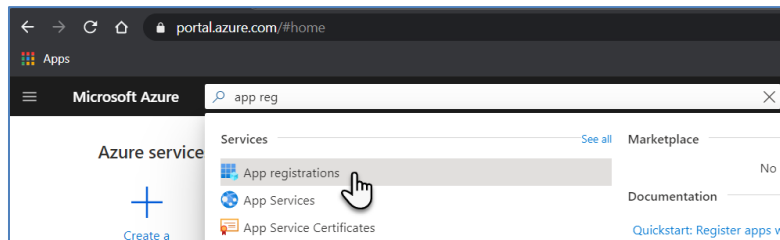
Retrieve data from MS Graph API

Before MS Graph API can be queried, permissions in Azure must be set.

A step by step instruction can be found on: <https://www.lee-ford.co.uk/getting-started-with-microsoft-graph-with-powershell/>

In short.

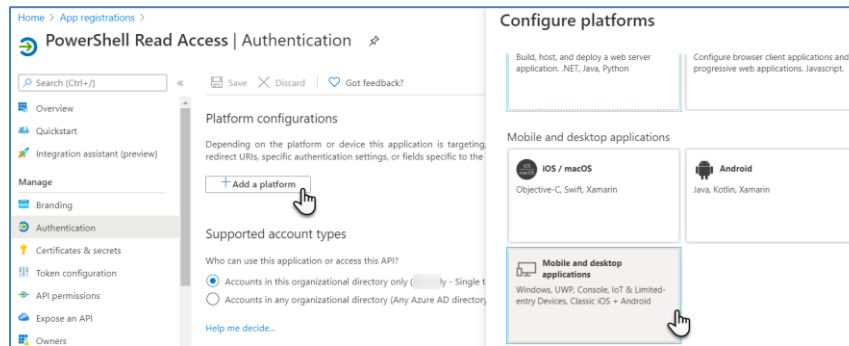
1. Navigate to **portal.azure.com**
2. Search for **app registrations** and click on it



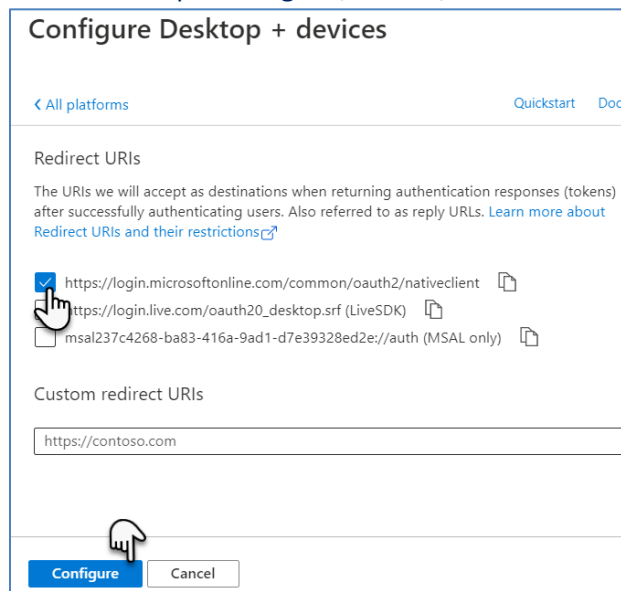
3. Click on **new registration**
4. Set a name, e.g. **PowerShell Read Access** and click on **Register**

A screenshot of the 'Register an application' form in the Azure portal. The form is titled 'Register an application' and has a breadcrumb trail 'Home > App registrations >'. The first section is labeled '* Name' and contains a text input field with the value 'PowerShell Read Access'. Below this, there is a section for 'Supported account types' with the heading 'Who can use this application or access this API?'. There are three radio button options: 'Accounts in this organizational directory only (Single tenant)', 'Accounts in any organizational directory (Any Azure AD directory - Multitenant)', and 'Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)'. The first option is selected. At the bottom of the form, there is a 'Register' button with a mouse cursor pointing at it. A link 'Help me choose...' is also visible.

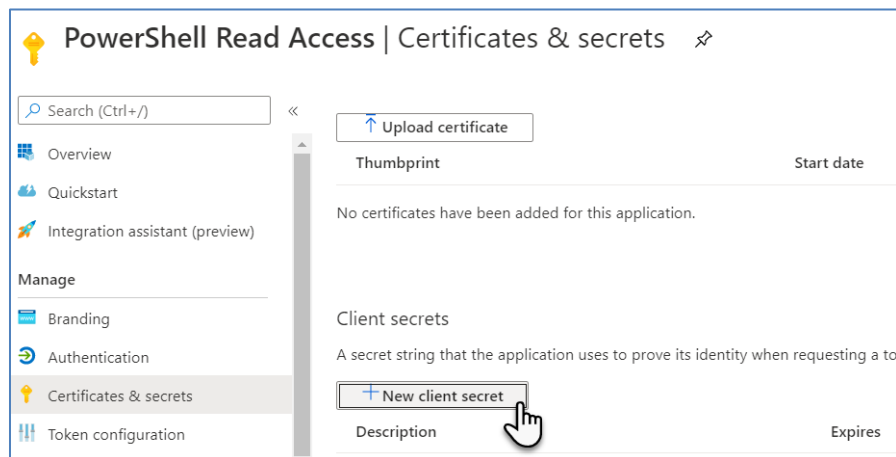
5. In **Authentication** select **Add a platform** and choose **Mobile and desktop applications**



6. Tick the first option **“login.. /oauth2/nativeclient”** and confirm with **configure**



7. In **Certificates & secrets** choose **New client secret**



8. Add a **description** and choose an **expiration time** and proceed with **Add**.

The screenshot shows the 'Add a client secret' dialog box. The 'Description' field is filled with 'QueryPass for Scripts'. The 'Expires' section has three radio buttons: 'In 1 year' (selected), 'In 2 years', and 'Never'. The 'Add' button is highlighted with a hand cursor.

9. Copy the password to the clipboard and store it into a Password-Safe!

The screenshot shows the 'Client secrets' page. It contains a table with one entry:

Description	Expires	Value
QueryPass for Scripts	7/22/2021	bc... \hj,j~z

A 'Copy to clipboard' button is visible next to the value field. A hand cursor is pointing at the copy icon.

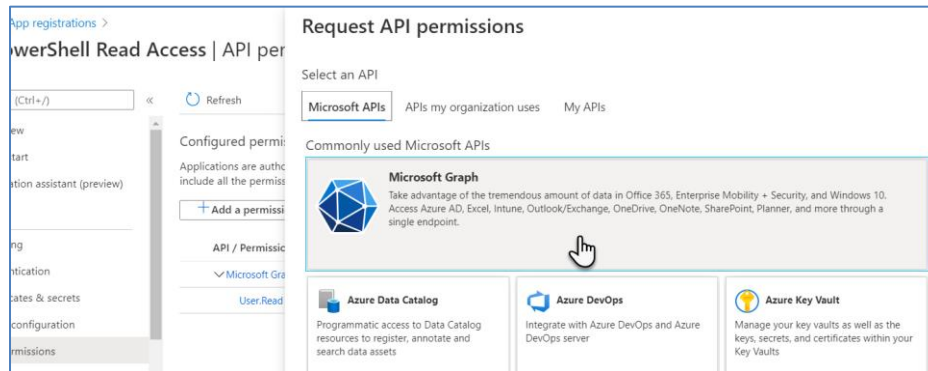
This password is the **clientSecret** and required when connecting with SCOM Task.

10. Change to **API permissions** and click on **Add permissions**

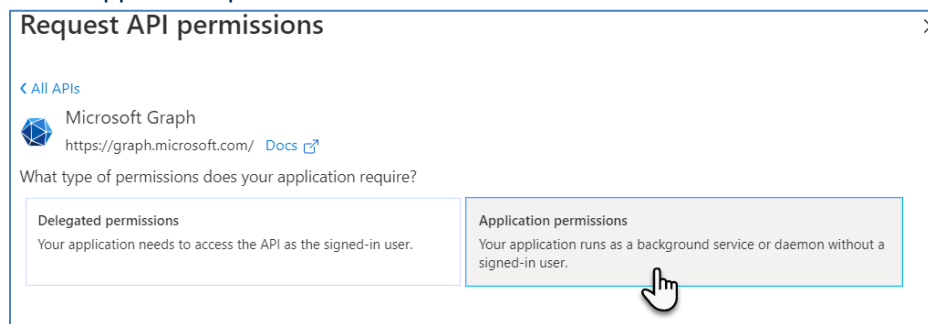
The screenshot shows the 'API permissions' page. The 'Add a permission' button is highlighted with a hand cursor. The table below shows the configured permissions:

API / Permissions name	Type	Description
Microsoft Graph (1)		
User.Read	Delegated	Sign in and read user profile

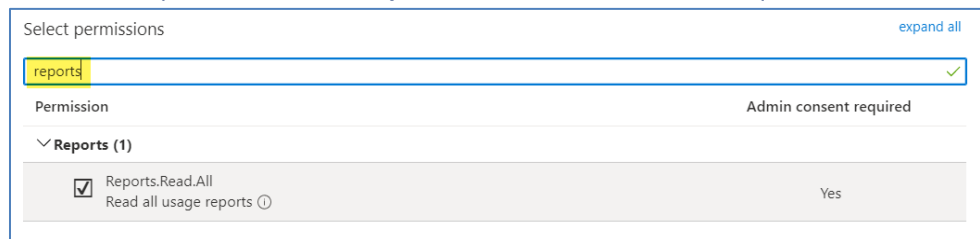
11. Select Microsoft Graph



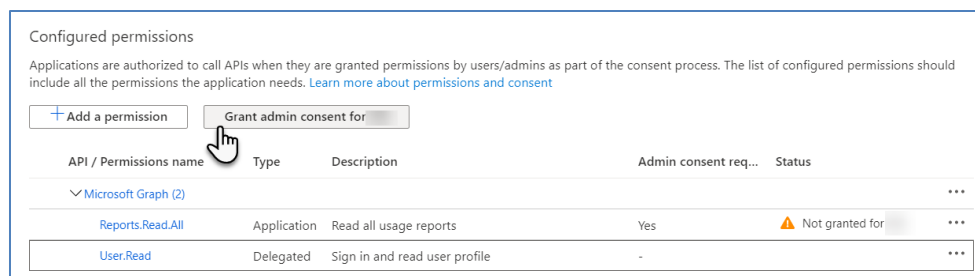
12. Select Application permissions



13. Search for reports and select **Reports.Read.All** and others if required.



14. Click on **Grant admin consent for ...**



afterwards it should look as below:

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 include all the permissions the application needs. [Learn more about permissions and consent](#)

Add a permission

Grant admin consent for

API / Permissions name	Type	Description	Admin consent req...	Status
▼ Microsoft Graph (2)				
Reports.Read.All	Application	Read all usage reports	Yes	<div><div></div></div> Granted for
User.Read	Delegated	Sign in and read user profile	-	<div><div></div></div> Granted for

15. Note down the **client ID (yellow)** and the **tenant ID (blue-ish)** for usage.

Home > App registrations >

PowerShell Read Access

Search (Ctrl+/)

Delete
Endpoints

Overview

Quickstart

Integration assistant (preview)

Manage

Branding

Authentication

Certificates & secrets

Got a second? We would love your feedback on Microsoft identity platform (previously Azure AD for developer). →

Display name

PowerShell Read Access

Application (client) ID

237c4268-ba83-416a-9ad1-d7e39328ed2e

Directory (tenant) ID

f0660551-7f49-4caa-9af7-c22f7cb31c24

Object ID

3f3d8e8f-f247-4faa-a641-d66f2b72bb82

Supported account types

My organization only

Redirect URIs

0 web, 0 spa, 1 public client

Application ID URI

Add an Application ID URI

Managed application in local directory

PowerShell Read Access

Retrieve data from MS O365 Management API

Based on the steps above, add the following API permissions:

Office 365 Management API, Application Permission, ServiceHealth READ and grant Admin consent. The result shall look like similar the screenshot below.

Certificates & secrets

Token configuration

API permissions

Expose an API

Owners

Roles and administrators (Preview)

Other permissions granted for

These permissions have been granted for but aren't in the configured permissions list. If your application requires these permissions, you should cons to the configured permissions list. [Learn more](#)

API / Permissions name	Type	Description	Admin consent required	Status
▼ Office 365 Management APIs (1)				
ServiceHealth.Read	Application	Read service health information for your org...	Yes	✔ Granted for

For additional information about Groups and SharePoint sites, consider these permissions, too:

API / Permissions name	Type	Description	Admin consent req...	Stat
Microsoft Graph (12)				
AuditLog.Read.All	Application	Read all audit log data	Yes	✔
Directory.Read.All	Application	Read directory data	Yes	✔
Files.Read.All	Application	Read files in all site collections	Yes	✔
Group.Read.All	Application	Read all groups	Yes	✔
Organization.Read.All	Application	Read organization information	Yes	✔
Reports.Read.All	Application	Read all usage reports	Yes	✔
Sites.Read.All	Application	Read items in all site collections (preview)	Yes	✔
User.Read	Delegated	Sign in and read user profile	-	✔
Office 365 Management APIs (1)				
ServiceHealth.Read	Application	Read service health information for your organization	Yes	✔