

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.

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

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
Preparation.....	8
Retrieve data from MS Graph API	8

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:



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:

Task Output

Copy Text
Copy HTML

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:

Run As:

Run Location:

Target:

Target Type:

Windows Computer

Category:

Operations

Task Description

Displays the computers' last changes; last installed software, logged on user, etc.
 Note: JSON format is only supported if PowerShell v3 or later is installed on the target server.

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:

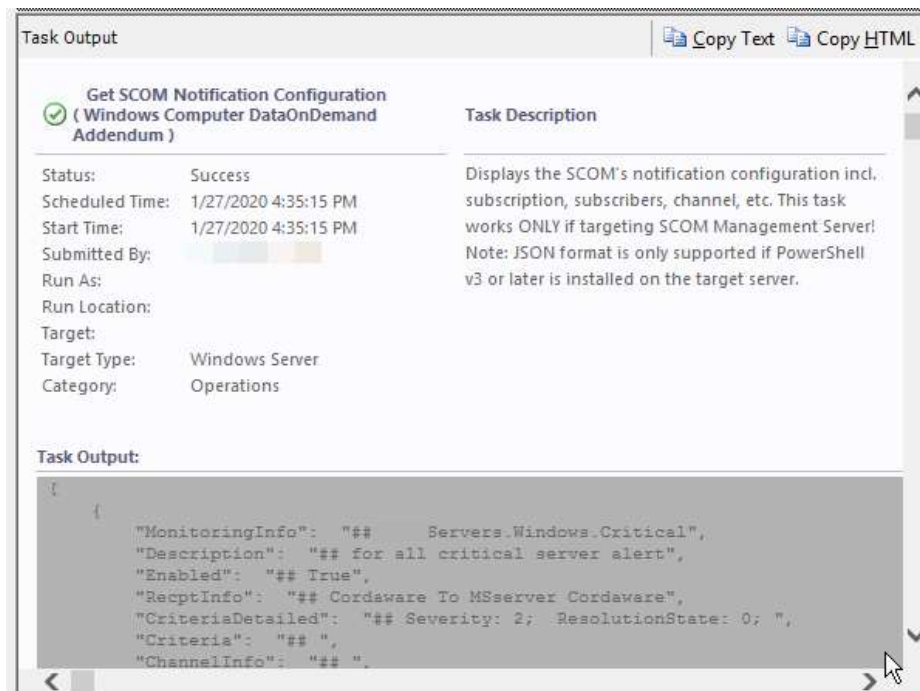
Displays the SCOM's notification configuration incl. subscription, subscribers, channel, etc.

→ 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 execution window titled "Task Output". It contains a summary of the task "Get-RemoteSOAPServiceInfo (Windows Computer DataOnDemand Addendum)" and its output. The task status is "Success". The output is a JSON array containing one object with various properties.

Task Description

Queries a remote SOAP service (XML) displays it's data. Note: JSON format is only supported if PowerShell v3 or later is installed on the target server. Information for Headers are not yet considered. Support will come in a later version.

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

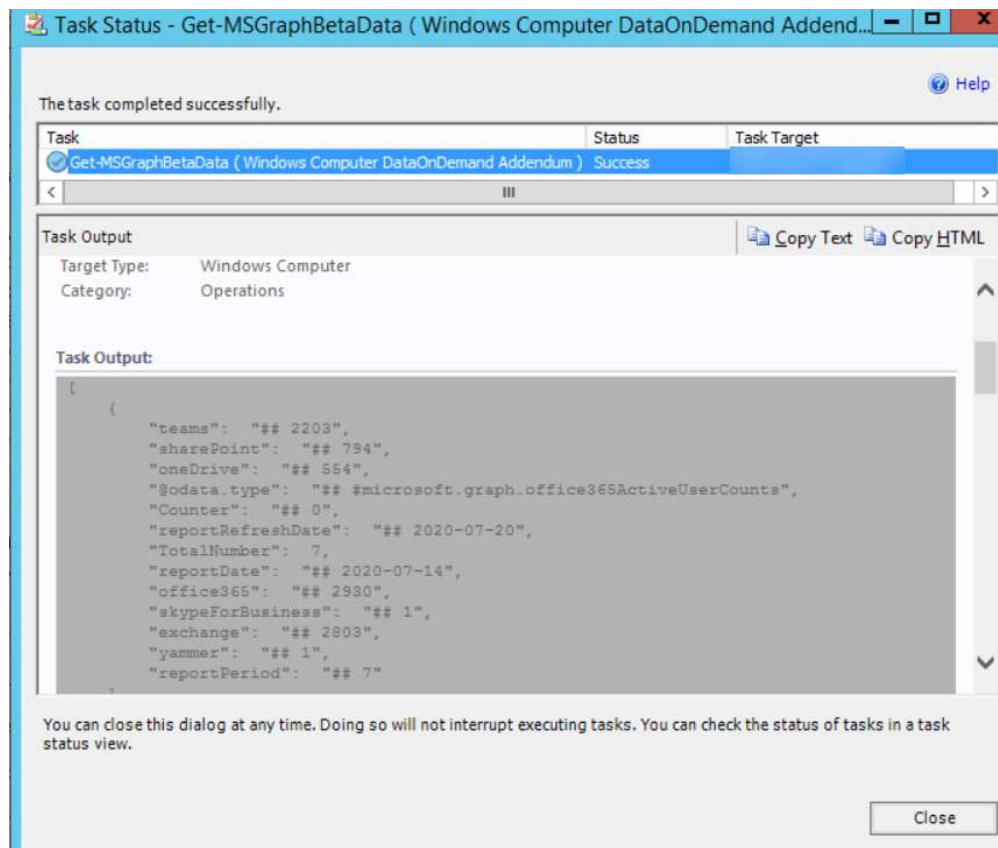
Description:

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:



Preparation

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

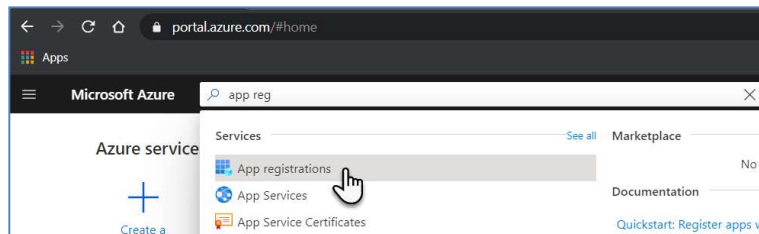
Retrieve data from MS Graph API

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

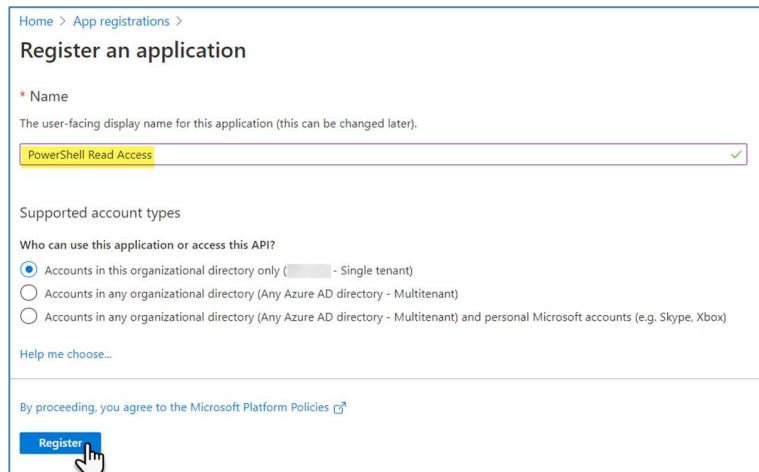
A step by step instruction can be found on: <https://www.lee-ford.co.uk/getting-started-with-microsoft-graph-with-powershell/>

In short.

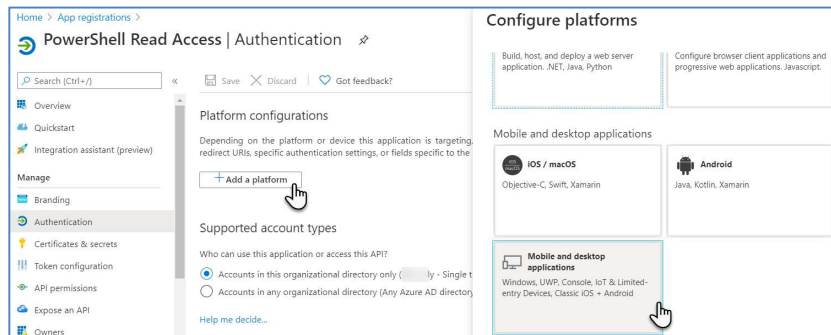
1. Navigate to **portal.azure.com**
2. Search for **app registrations** and click on it



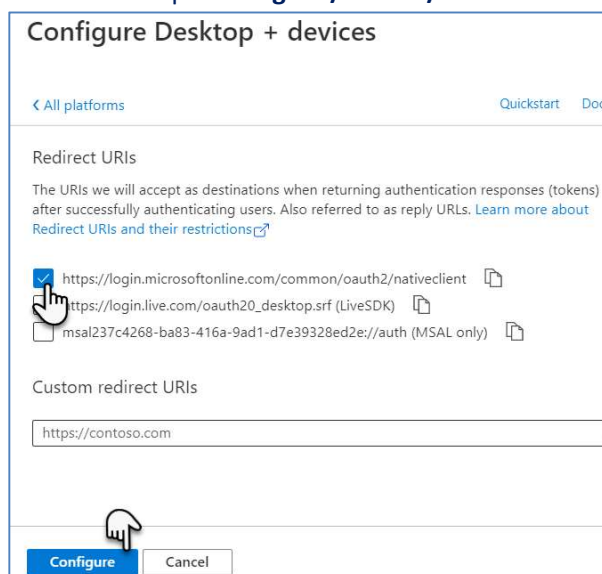
3. Click on **new registration**
4. Set a name, e.g. **PowerShell Read Access** and click on **Register**

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 'Name', with a text input field containing 'PowerShell Read Access' and a green checkmark icon. Below this is the 'Supported account types' section, which asks 'Who can use this application or access this API?'. There are three radio button options: 'Accounts in this organizational directory only (Single tenant)' (selected), '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)'. At the bottom of the form, there is a checkbox for 'By proceeding, you agree to the Microsoft Platform Policies' and a blue 'Register' button with a mouse cursor hovering over it.

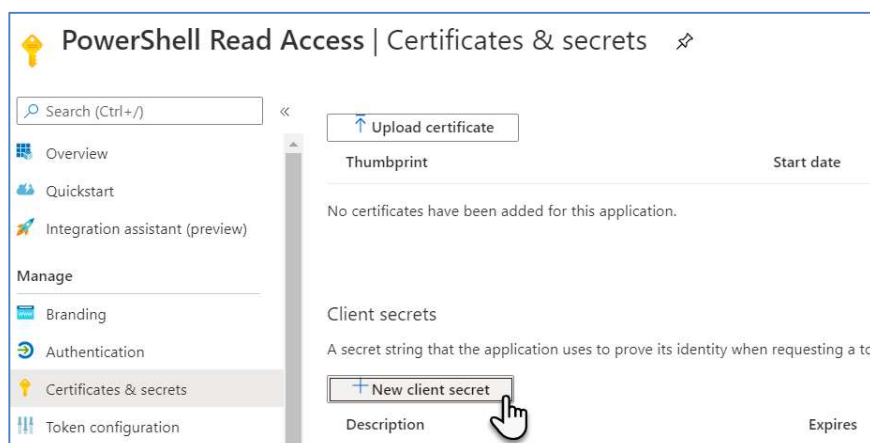
5. In **Authentication** select **Add a platform** and choose **Mobile and desktop applications**



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**.

PowerShell Read Access | Certificates & secrets

Search (Ctrl+ /)

Overview

Quickstart

Integration assistant (preview)

Age

Branding

Authentication

Certificates & secrets

Add a client secret

Description

QueryPass for Scripts

Expires

☒ In 1 year

☐ In 2 years

☐ Never

Add Cancel

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

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
QueryPass for Scripts	7/22/2021	b6...hj_J~z

Copy to clipboard

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

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

PowerShell Read Access | API permissions

Search (Ctrl+ /)

Refresh

Overview

Quickstart

Integration assistant (preview)

Manage

Branding

Authentication

Certificates & secrets

Token configuration

API permissions

Configured permissions

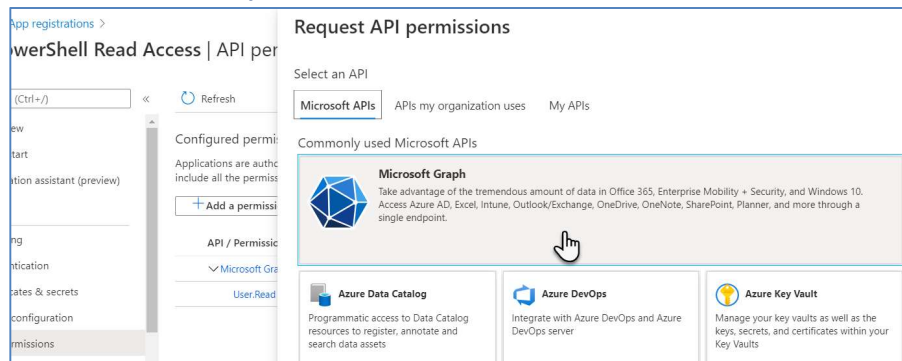
Applications are authorized to call APIs when they are granted permissions by users/administrators. Include all the permissions the application needs. [Learn more about permissions and consent](#)

+ Add a permission

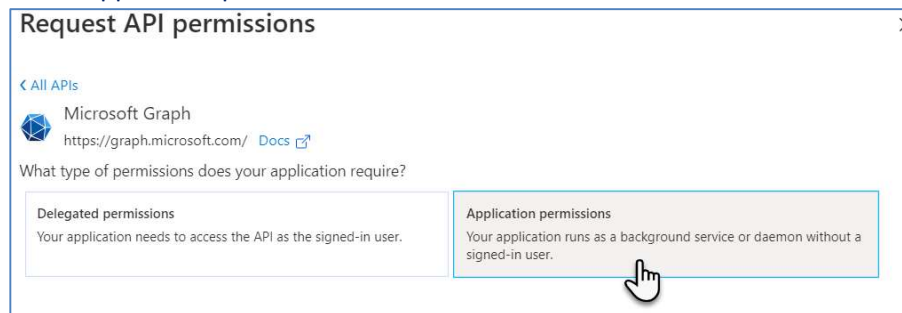
Grant admin consent for

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

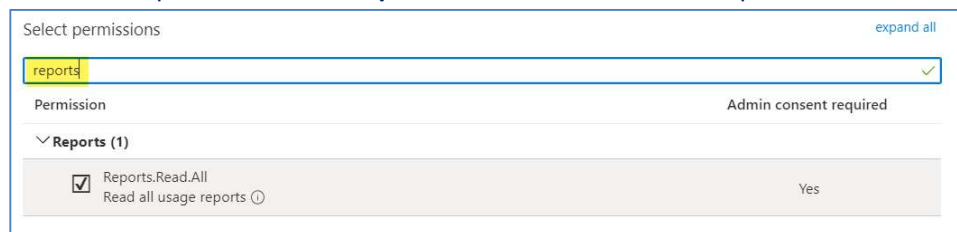
11. Select Microsoft Graph



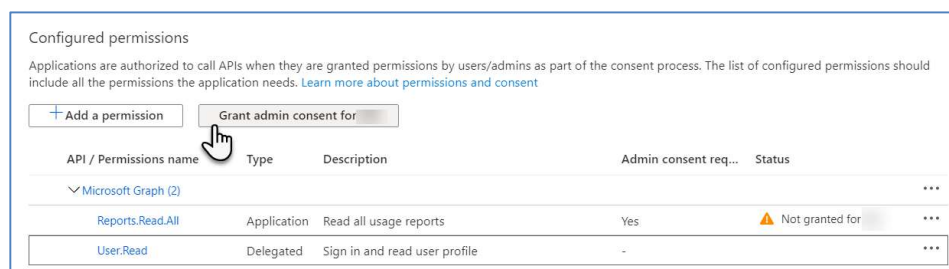
12. Select Application permissions



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

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

Home > App registrations >

PowerShell Read Access

Search (Ctrl+J)

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

0660551-7f49-4caa-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