Read.md

**Introduction:**

Entra ID Entitlement Management using Microsoft Graph PowerShell enables you to manage access to all the resources that users need, resources like groups, applications, sites etc. Entitlement Management helps to create a package of resources that internal users can use for self-service requests. Requests could be Multi-level, Single-Level, Auto-Assigned (Query Based) or Admin Managed (Query based).

Here, in this script, requestors would get membership of resources belonging to catalog after successfully execution of defined policy jobs.

*Note: These resources could be groups, applications or SharePoint Online sites.*

**Requirements:**

1. Entra ID tenant with one of the licenses to use “Entra ID Entitlement Management”
   1. Azure AD Premium P2. OR
   2. Enterprise Mobility + Security (EMS) E5 license.
2. Microsoft Graph PowerShell SDK.
3. “User Administrator”, “Identity Governance Administrator” or “Global Administrator” PIM role is required to configure catalogs, access packages or policies in entitlement management.
4. Entra ID Groups to onboard as a catalog resource and Access package members.

This PowerShell script can be utilized by the Application Registration Service Principal which should have permissions mentioned below:

|  |  |  |  |
| --- | --- | --- | --- |
| **API/Permission Name** | **Type** | **Description** | **Admin Consent Required** |
| Application.Read.All | Application | Read all applications | Yes |
| EntitlementManagement.ReadWrite.All | Application | Read and write entitlement management resources. | Yes |
| Group.ReadWrite.All | Application | Read and write all groups | Yes |
| User.Read.All | Application | Read all users full profiles. | Yes |

**Use Case:**

This Script will consume Csv file as an input, where delimitation based on “tab” (`t), which contains below defined attributes along with their roles.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute Name** | **Type** | **Remarks** | **Mandatory** |
| Catalog\_Name | String | Name of catalog | Yes |
| Catalog\_Desc | String | Description for catalog | Optional |
| GroupResources | Array | List of groups segregated based on semicolon (;) | Yes |
| APName | String | Access Package Name | Yes |
| APDesc | String | Description for access policy. | Optional |
| APPolicyName | String | Access Package Policy Name | Yes |
| APPolicyDesc | String | Description for Access Package policy. | Optional |
| Approval level | String | Currently the script has 4 cases i.e., Multi-Level, Single-Level, Auto-Assigned & others | To be defined based on requirement. |
| Filter | String | In the case of Auto-Assigned, there must be some criteria (Regular Expression format) to be defined to auto-assign the package to end user. | Mandatory to be defined in case of Auto-Assigned. |
| Approver1 | String | Values can be considered for **Approver1** is either of 1 value between Group Name, User’s UserPrincipalName value or “Manager”. | Yes (If Applicable) |
| Approver2 | String | Values can be considered for **Approver1** is either of 1 value between Group Name, User’s UserPrincipalName value or “Manager”.  ***Note 1***: It is recommended we do not assign the same object as Approver 2 as Approver 1 to avoid unexpected outcomes.  ***Note 2:*** This should have value only Multi-Level Approval case, leave it blank if not applicable. | Yes (If Applicable) |
| ReqAutoDenialDuration | String | Duration after which request would be considered as “**Denial**” if there is no response from approver. | Yes |
| FallBackApprover | String | Values can be considered for **FallBackApprover** is either of 1 value between Group Name, User’s UserPrincipalName value. | Yes (only in case of multi-level or single-level approval) |
| Reviewer | String | Values can be considered for **Reviewer** is either of 1 value between Group Name, User’s UserPrincipalName value. | Yes (only in case of multi-level or single-level approval) |

There are some assumptions and considerations while using this script in environments.

1. Once the Catalog is generated, Script will generate 4 associated groups for each catalog based on Roles and responsibilities, which can be utilized by Admin to assign user to manage catalog and access package and their assignments to segregate the administrative roles to business owners to provide flexibility to manage by their own. These roles are mentioned below:

| **Entitlement management role** | **Role definition ID** | **Description** |
| --- | --- | --- |
| Catalog owner | ae79f266-94d4-4dab-b730-feca7e132178 | Edit and manage access packages and other resources in a catalog. Typically an IT administrator or resource owners, or a user who the catalog owner chooses. |
| Catalog reader | 44272f93-9762-48e8-af59-1b5351b1d6b3 | View existing access packages within a catalog. |
| Access package manager | 7f480852-ebdc-47d4-87de-0d8498384a83 | Edit and manage all existing access packages within a catalog. |
| Access package assignment manager | e2182095-804a-4656-ae11-64734e9b7ae5 | Edit and manage all existing access packages' assignments. |

1. After the completion of script, Eligible users are allowed to request access using <https://myaccess.microsoft.com/> by selecting recently created catalog, which will follow the approval process, and once it is approved or request processed, group membership would be assigned to the requestor.
2. During execution of script, App SPN will be assigned as an **owner** of the resources belongs to catalog to allow script to assign role in the resource in Access package to avoid error "*The caller is not the resource owner”.*

**Note: *Purpose of this script is to simulate the specific requirement, to utilize the script please validate the parameters and change the script based on business requirements.***