

RBAC setup to use Graph Service in Service Suite

RBAC (Role-Based Access Control) for applications will eventually replace application access policies (see below). Today, the public preview shows some ragged edges in terms of the integration between Exchange Online and Azure AD that Microsoft says they'll fix before general availability. Even with some bumps to navigate, the new functionality is a welcome addition that extends tenant control over their data, which is always a good thing.

In order to use this protocol you need to do the following configurations in your azure account so the roles and permissions are set correctly.

Once you have done the below steps you should be able to use the service suite Graph Service oauth2 client to send/receive emails using RBAC with the normal client commands.

1. Open Windows PowerShell as an Administrator
2. Execute: `connect-mggraph -scopes "Application.Read.All"`. If the exception "*The term connect-mggraph' is not recognized as the name of a cmdlet*" appears, execute the command

```
install-module microsoft.graph
```

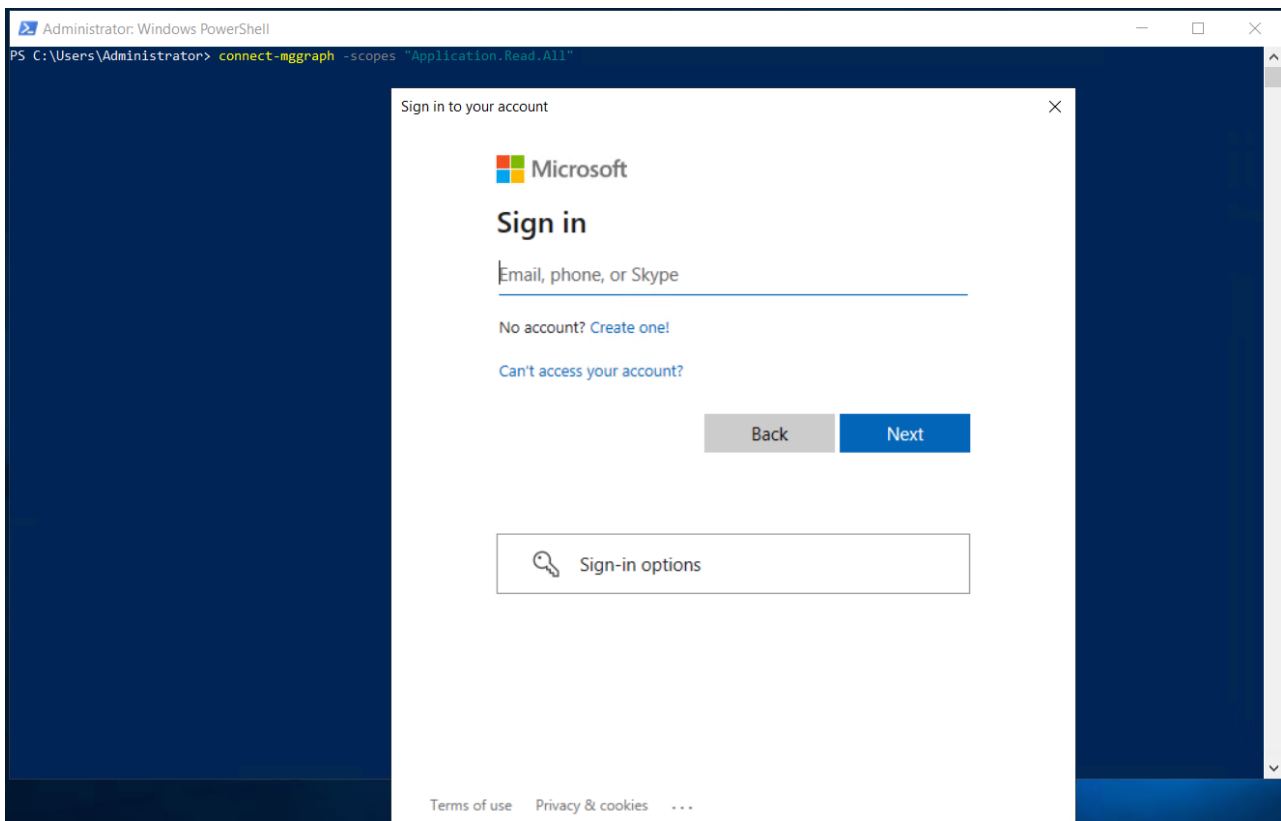
```
PS C:\Users\Administrator> install-module microsoft.graph

NuGet provider is required to continue
PowerShellGet requires NuGet provider version '2.8.5.201' or newer to interact with NuGet-based repositories. The NuGet provider must be available in 'C:\Program Files\PackageManagement\ProviderAssemblies' or 'C:\Users\Administrator\AppData\Local\PackageManagement\ProviderAssemblies'. You can also install the NuGet provider by running 'Install-PackageProvider -Name NuGet -MinimumVersion 2.8.5.201 -Force'. Do you want PowerShellGet to install and import the NuGet provider now?

[Y] Yes [N] No [S] Suspend [?] Help (default is "Y"): y

Untrusted repository
You are installing the modules from an untrusted repository. If you trust this repository, change its InstallationPolicy value by running the Set-PSRepository cmdlet. Are you sure you want to install the modules from 'PSGallery'?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): a
PS C:\Users\Administrator>
```

3. After Microsoft Graph is installed, execute again the command `connect-mggraph -scopes "Application.Read.All"`
4. Sign in



5. After signing in, should see this:

```
PS C:\WINDOWS\system32> connect-mggraph -scopes "Application.Read.All"
Welcome To Microsoft Graph!
```

6. Execute: `$SP = Get-MgServicePrincipal -All`

7. Execute: `$ServicePrincipalId = $SP | Where-Object {$_.displayName -eq "{Application name}"} | Select-Object -ExpandProperty Id`

8. Execute: `$AppId = $SP | Where-Object {$_.displayName -eq "{Application name}"} | Select-Object -ExpandProperty AppId`

9. Execute: `Write-Host ("AppId is {0} and Service Principal Id is {1}" -f $AppId, $ServicePrincipalId)`

Output:

```
PS C:\WINDOWS\system32> Write-Host ("AppId is {0} and Service Principal Id is {1}" -f $AppId, $ServicePrincipalId)
AppId is a08a1401 and Service Principal Id is c8280bc6
```

10. Execute: `Install-Module -Name ExchangeOnlineManagement`

11. Execute: `Import-module ExchangeOnlineManagement`

12. Execute: `Connect-ExchangeOnline -UserPrincipalName {user email}`

Output:

```
PS C:\WINDOWS\system32> Connect-ExchangeOnline -UserPrincipalName kjimenez@appvance.com

-----
This V3 EXO PowerShell module contains new REST API backed Exchange Online cmdlets which doesn't require WinRM for Client-Server communication. You can now run these cmdlets after turning off WinRM Basic Auth in your client machine thus making it more secure.

Unlike the EXO* prefixed cmdlets, the cmdlets in this module support full functional parity with the RPS (V1) cmdlets.

V3 cmdlets in the downloaded module are resilient to transient failures, handling retries and throttling errors inherently.

However, REST backed EOP and SCC cmdlets are not available yet. To use those, you will need to enable WinRM Basic Auth.

For more information check https://aka.ms/exov3-module
-----
```

13. Execute: `New-ServicePrincipal -AppId $AppId -ServiceId $ServicePrincipalId -DisplayName '{Application name}'`

14. Execute: `New-ManagementScope -Name "Management emails" -RecipientRestrictionFilter "CustomAttribute1 -eq '{RolName}'"`

15. Execute: `Set-Mailbox -Identity {username} -CustomAttribute1 "{RolName}"`

16. `Execute: Get-EXOMailbox -Filter {CustomAttribute1 -eq "{RolName}"} | Format-Table DisplayName`
17. `Execute: New-ManagementRoleAssignment -App $AppId -Role "Application Mail.Mail.ReadWrite" -CustomResourceScope "Management emails"`
18. `Execute: New-ManagementRoleAssignment -App $AppId -Role "Application Mail.Send" -CustomResourceScope "Management emails"`

Assing a role to a user

`Execute: Set-Mailbox -Identity {username} -CustomAttribute1 "{RolName}"`

List available permissions

`Execute: Get-ManagementRole | Where-Object {$_.Name -like "Application *"}`

```
PS C:\WINDOWS\system32> Get-ManagementRole | Where-Object {$_.Name -like "Application *"}

Name                                     RoleType
----
Application Mail.Read                  ApplicationMailRead
Application Mail.ReadBasic             ApplicationMailReadBasic
Application Mail.ReadWrite             ApplicationMailReadWrite
Application Mail.Send                  ApplicationMailSend
Application MailboxSettings.Read       ApplicationMailboxSettingsRead
Application MailboxSettings.ReadWrite ApplicationMailboxSettingsReadWrite
Application Calendars.Read             ApplicationCalendarsRead
Application Calendars.ReadWrite        ApplicationCalendarsReadWrite
Application Contacts.Read              ApplicationContactsRead
Application Contacts.ReadWrite         ApplicationContactsReadWrite
Application Mail Full Access           MailFullAccessApp
Application Exchange Full Access      ExchangeFullAccessApp
Application EWS.AccessAsApp            ApplicationEWSAccessAsApp
```

See permissions assigned to a user

`Execute: Test-ServicePrincipalAuthorization -Identity $AppId -Resource {email}`

```
PS C:\Users\Administrator> Test-ServicePrincipalAuthorization -Identity $AppId -Resource kjimenez@appvance.com

RoleName      GrantedPermissions      ScopeType      InScope      AllowedResourceScope
-----      -
Application Mail.Read      Mail.Read      CustomRecipientScope      True      Management emails
Application Mail.Send      Mail.Send      CustomRecipientScope      True      Management emails
Application Mail.ReadWrite      Mail.ReadWrite      CustomRecipientScope      True      Management emails
```

Validation checklist

After assigning the Graph application roles, verify the service principal and application identifiers returned by PowerShell, then confirm that the OAuth2 client used by Service Suite has the required permissions. Test with a non-production mailbox first and record the tenant, application, and role assignments used for troubleshooting.

Last updated: October 15, 2026

If you have any feedback on this topic, please contact us at documentation@appvance.ai

Revision #3
Created 2026-07-31 15:39:36 UTC by Admin
Updated 2026-10-15 18:20:09 UTC by mmora