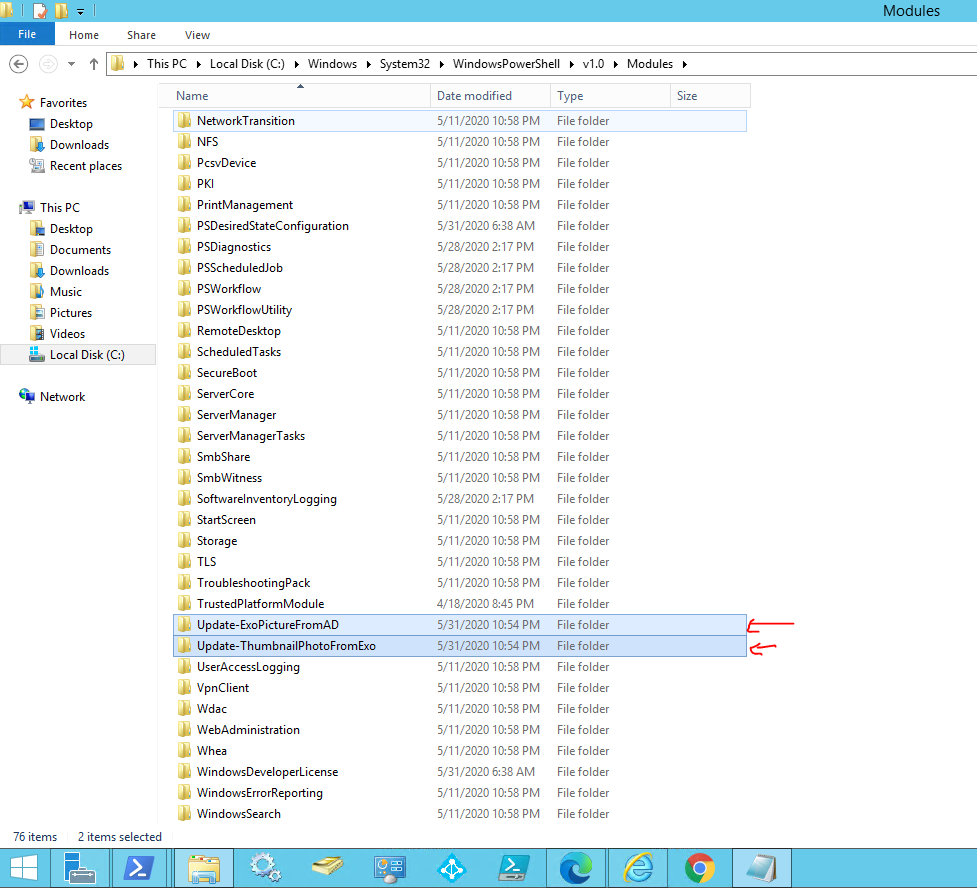
**This is a Documentation of how to Use the modules below:**

***Update-ExoPictureFromAD***

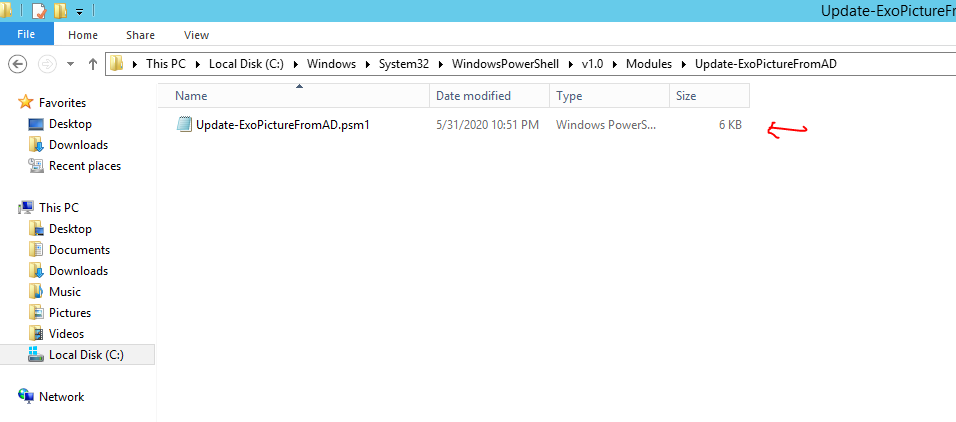
***Update-ThumbnailPhotoFromExo***

1. First Navigate to this location on your PC ***C:\Windows\System32\WindowsPowerShell\v1.0\Modules***

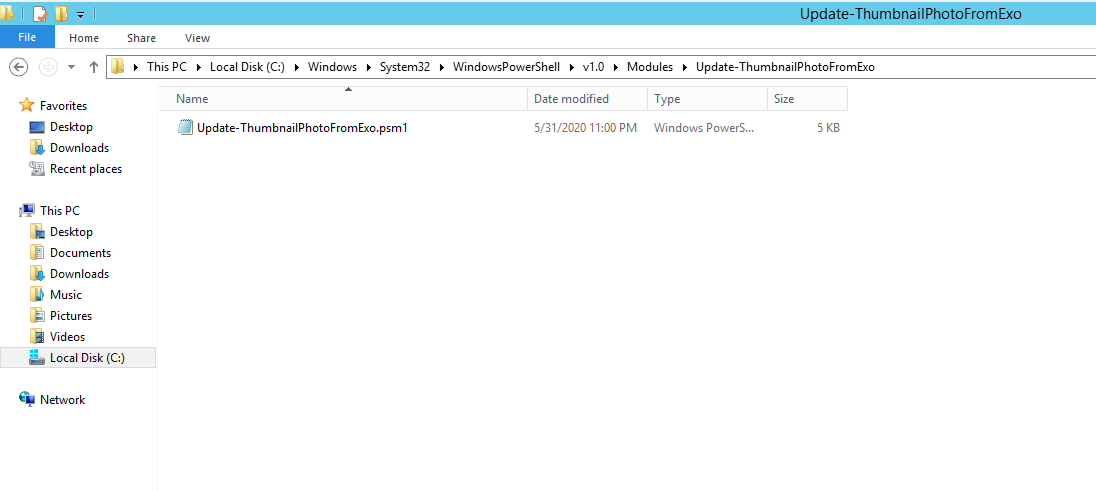
2. Create two Folders called ***Update-ExoPictureFromAD*** and ***Update-ThumbnailPhotoFromExo***



3. Now Copy the respective ***Update-ExoPictureFromAD.psm1*** file to the ***Update-ExoPictureFromAD*** folder



4. And Copy the respective ***Update-ThumbnailPhotoFromExo.psm1*** file to the ***Update-ThumbnailPhotoFromExo*** folder



***Update-ExoPictureFromAD*** is a function that (push/uploads) (pictures/thumbnailPhoto) attributes from AD to Exchange Online

***Update-ThumbnailPhotoFromExo*** is a function that pulls/uploads) (pictures/thumbnailPhoto) attributes from Exchange Online to AD thumbnailPhoto attribute.

**Now for modifications I have made, the Script should look like this:**

***Update-ExoPictureFromAD*** ***-First 0 -Last 1000***

The cmdlet above updates pictures for users on Exchange Online Ranging from 1 to 1000 in respect to the AD Users

**Note:** So, if you have 10,000 AD Users in the organization updates will be done/attempted for the first 1,000 AD Users to their respective Exchange Online Mailboxes

***Update-ThumbnailPhotoFromExo*** ***-First 0 -Last 1000***

The cmdlet above updates pictures/thumnailPhoto property for users on AD Ranging from 1 to 1,000 in respect to the Mailboxes in Exchange Online

**Note:** So, if you have 10,000 Exchange Online mailboxes in the organization updates will be done/attempted for the first 1,000 Exchange Online mailboxes to their respective AD Users

***Update-ExoPictureFromAD -FN 0 -LN 1000***

***Update-ThumbnailPhotoFromExo -FN 0 -LN 1000***

The cmdlet above does the same function as the previous one.

Only difference is its making use of Alias

**Note:** -FN is an alias for First

**Note:** -LN is an alias for Last

***Update-ThumbnailPhotoFromExo -First 200***

The cmdlet above updates pictures for users on Exchange Online Ranging from 200 AD Users to Last amount of AD User *(Total Users on AD)*

**Note:** if ***-First*** is the only parameter specified it will automatically use last amount of the AD User as the ***-Last*** parameter to run

So, if you have 10,000 mailboxes in Exchange Online it will skip updates for 0 to 199 AD Users and start updates for 200 to the 10,000 AD User

Same goes for ***Update-ExoPictureFromAD -First 200***

***Update-ThumbnailPhotoFromExo -Last 500***

The cmdlet above updates pictures for users on Exchange Online Ranging from m 0 to 500 AD Users

**Note:** So, if there are 10,000 mailboxes in Exchange Online it will only make updates for 0 to 500 AD Users and skip the rest (501 - 10000) AD Users

Same goes for ***Update-ExoPictureFromAD -Last 500***

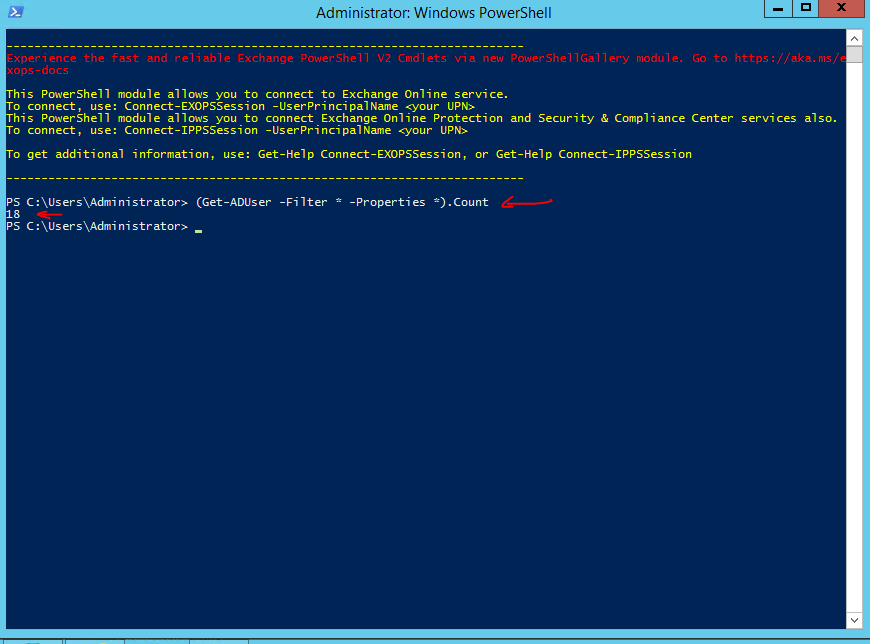
**Note:** For ***Update-ExoPictureFromAD*** ***-First 0 -Last 1000***

The Total size that should be taken into reference is the size of the all the AD users

Cmdlet below can be used to check

***(Get-ADUser -Filter \* -Properties \*).Count***

Screen shot below shows I have 18 on AD users



If I were to run for the cmdlet ***Update-ExoPictureFromAD*** for the first 5 AD users to try to upload their thumbnail property to Exchange Online, it will look like this

***Update-ExoPictureFromAD*** ***-First 0 -Last 5***

If I were to run it for the next 6 to 15 users, it will look like this

***Update-ExoPictureFromAD*** ***-First 6 -Last 15***

Then if it was for the last 3 users it will look like this

***Update-ExoPictureFromAD*** ***-First 16 -Last 18***

**OR**

***Update-ExoPictureFromAD*** ***-First 16***

*Remember if the last value is not entered it automatically picks the Total number of AD users as the last value*

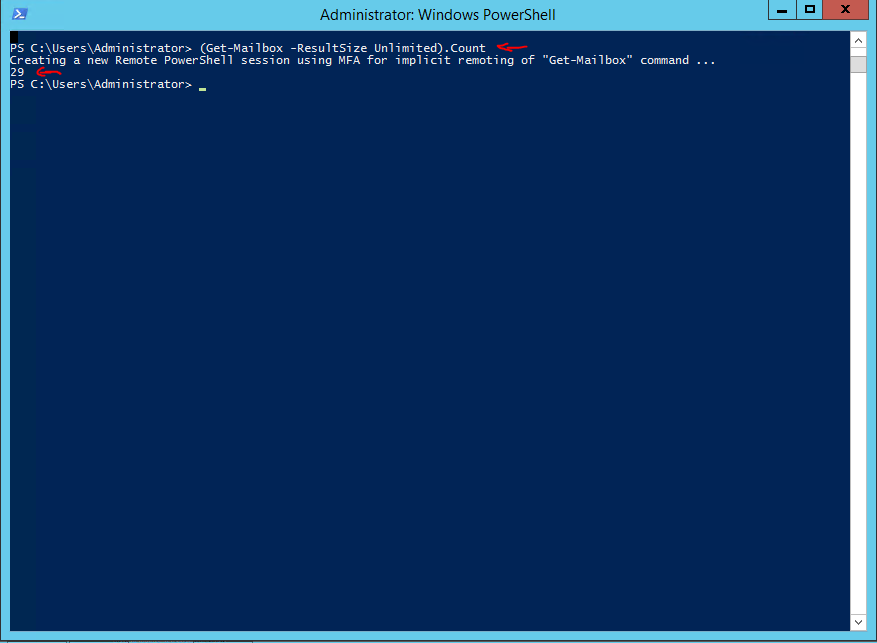
*Total number of AD users =* ***(Get-ADUser -Filter \* -Properties \*).Count***

While For ***Update- ThumbnailPhotoFromExo*** ***-First 0 -Last 1000***

The Total size that should be taken into reference is the size of the all the Mailboxes on Exchange online

***(Get-Mailbox -ResultSize Unlimited).Count***

Screen shot below shows I have a Total of 29 Mailboxes in the cloud



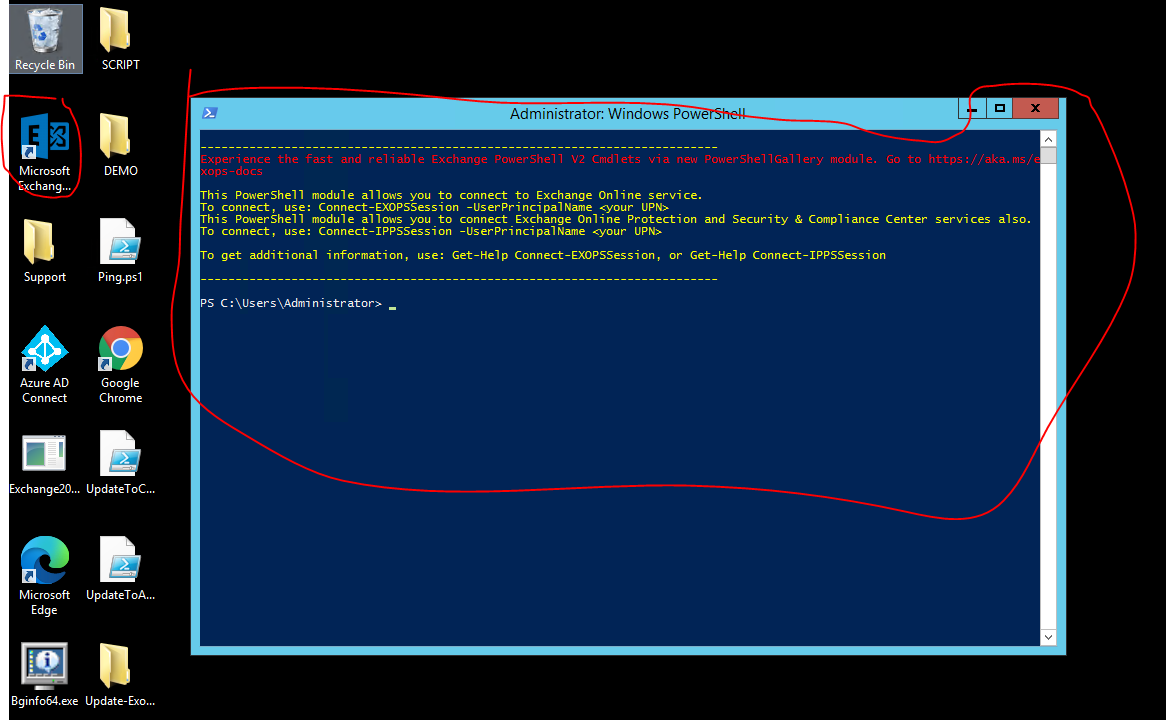
**NOTE:** if ***-Last*** is not specified the number 29 is picked automatically

***KINDLY NOTE:*** Logs are available on the desktop folder location on Server where Script is running from.

**TRYING TO USE TASK SCHEDULER WITH THIS**

I noticed that you might not be able to use the Exchange online Hybrid Module to make Automation with the Task Scheduler.

Image of the Exchange Hybrid Module is below



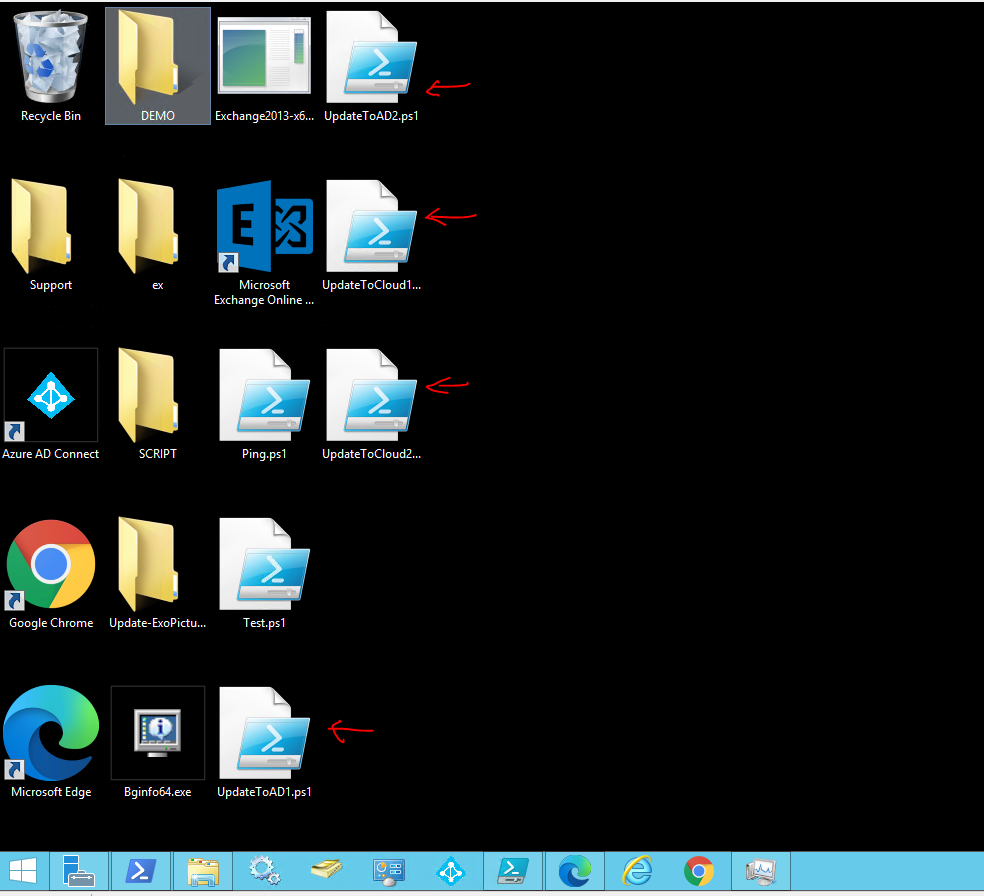
TRY looking into the Use the Exchange Online PowerShell V2 module if you are making use of MFA or Modern Authentication since the Exchange Hybrid Module is not compactible with running automated process

**Link as guide:** <https://docs.microsoft.com/en-us/powershell/exchange/exchange-online/exchange-online-powershell-v2/exchange-online-powershell-v2?view=exchange-ps>

I will be using the V2 module for demonstration

I have created 4 files on my Desktop called

1. UpdateToCloud1.ps1
2. UpdateToAD1.ps1
3. UpdateToCloud2.ps1
4. UpdateToAD2.ps1



The contents of ***UpdateToCloud1.ps1*** is as below:

Import-Module Update-ExoPictureFromAD

$uname = "username@domain.onmicrosoft.com"

$pass = "\*\*\*\*\*\*\*\*\*"

$password = ConvertTo-SecureString $pass -AsPlainText -Force

$UserCredential = New-Object System.Management.Automation.PSCredential ($uname,$password)

Connect-ExchangeOnline -Credential $UserCredential -ShowProgress $true

Update-ExoPictureFromAD -First 0 -Last 16

Disconnect-ExchangeOnline -Confirm:$false

The contents of ***UpdateToCloud2.ps1*** is as below:

Import-Module Update-ExoPictureFromAD

$uname = "username@domain.onmicrosoft.com"

$pass = "\*\*\*\*\*\*\*\*\*"

$password = ConvertTo-SecureString $pass -AsPlainText -Force

$UserCredential = New-Object System.Management.Automation.PSCredential ($uname,$password)

Connect-ExchangeOnline -Credential $UserCredential -ShowProgress $true

Update-ExoPictureFromAD -First 17 -Last 18

Disconnect-ExchangeOnline -Confirm:$false

The contents of ***UpdateToAD1.ps1*** is as below:

Import-Module Update-ThumbnailPhotoFromExo

$uname = "username@domain.onmicrosoft.com"

$pass = "\*\*\*\*\*\*\*\*\*"

$password = ConvertTo-SecureString $pass -AsPlainText -Force

$UserCredential = New-Object System.Management.Automation.PSCredential ($uname,$password)

Connect-ExchangeOnline -Credential $UserCredential -ShowProgress $true

Update-ThumbnailPhotoFromExo -First 0 -Last 25

Disconnect-ExchangeOnline -Confirm:$false

The contents of ***UpdateToAD2.ps1*** is as below:

Import-Module Update-ThumbnailPhotoFromExo

$uname = "username@domain.onmicrosoft.com"

$pass = "\*\*\*\*\*\*\*\*\*"

$password = ConvertTo-SecureString $pass -AsPlainText -Force

$UserCredential = New-Object System.Management.Automation.PSCredential ($uname,$password)

Connect-ExchangeOnline -Credential $UserCredential -ShowProgress $true

Update-ThumbnailPhotoFromExo -First 26 -Last 29

Disconnect-ExchangeOnline -Confirm:$false

I would be explaining section of one of the contents of the PS1 files I created above

#this imports the module

Import-Module Update-ThumbnailPhotoFromExo

#service account username or global admin username

$uname = "username@domain.onmicrosoft.com"

#password entered as string

$pass = "\*\*\*\*\*\*\*\*\*"

#password converted to secure string

$password = ConvertTo-SecureString $pass -AsPlainText -Force

#created an object for automation of credential with C# class

$UserCredential = New-Object System.Management.Automation.PSCredential ($uname,$password)

#Use the V2 module to connect to Exchange Online for users without MFA

Connect-ExchangeOnline -Credential $UserCredential -ShowProgress $true

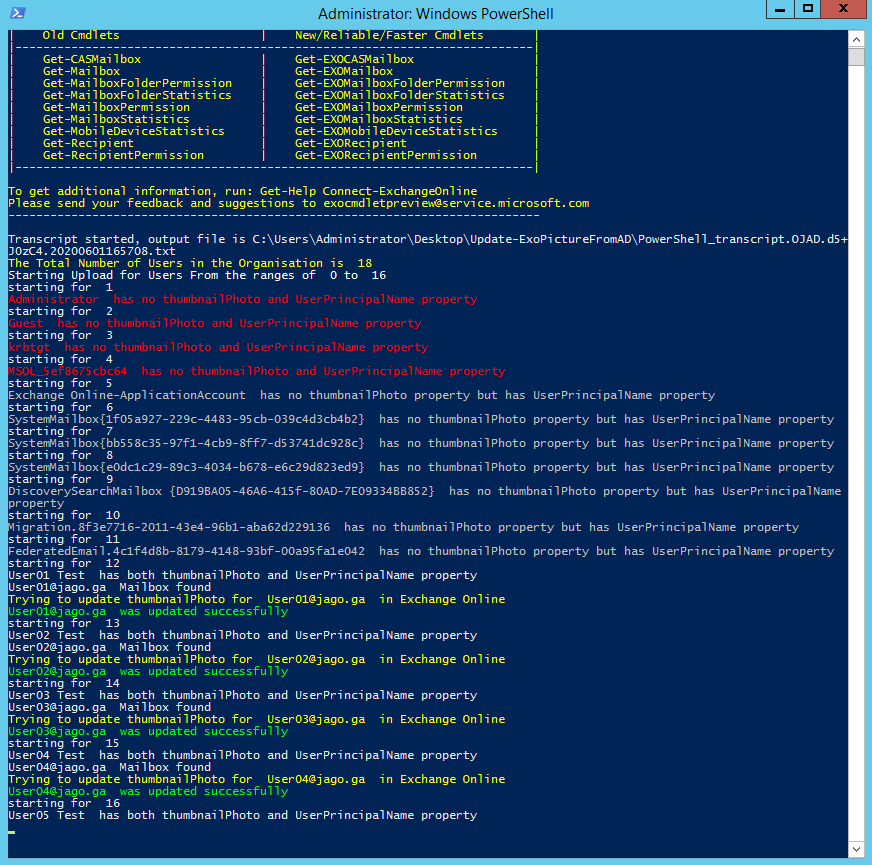
#Calling built in module

Update-ThumbnailPhotoFromExo -First 26 -Last 29

#Disconnecting from Exchange online V2 module --- a must for security reasons

Disconnect-ExchangeOnline -Confirm:$false

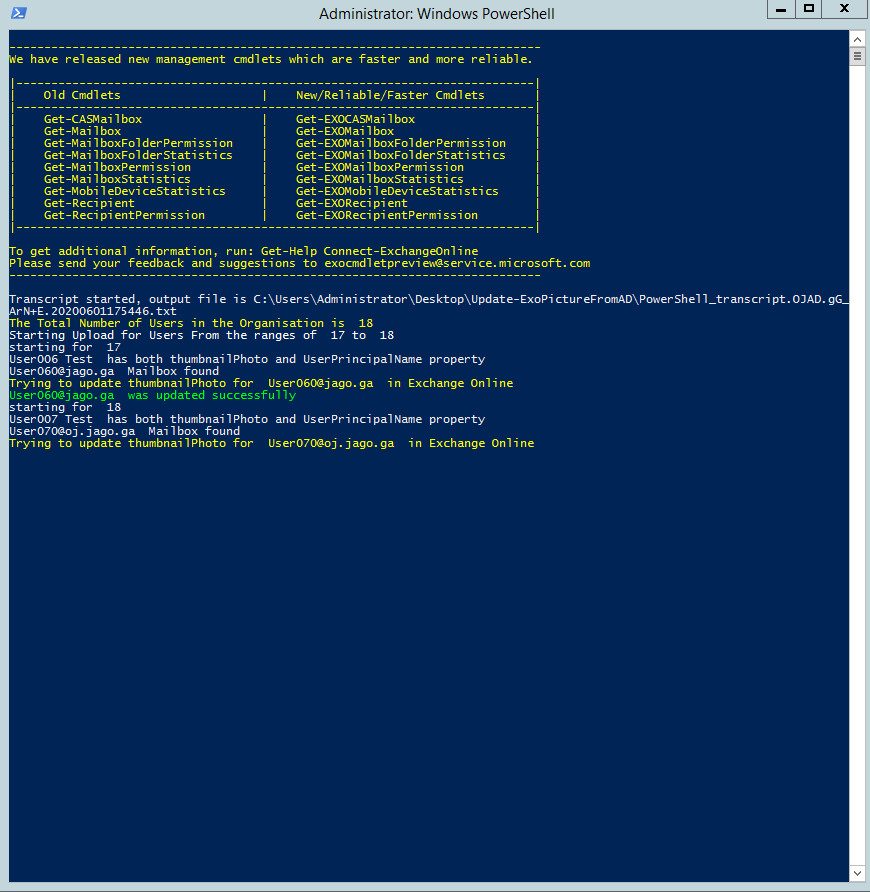
Screenshot of Output from the result of ***UpdateToCloud1.ps1***



Log File from the Above Result



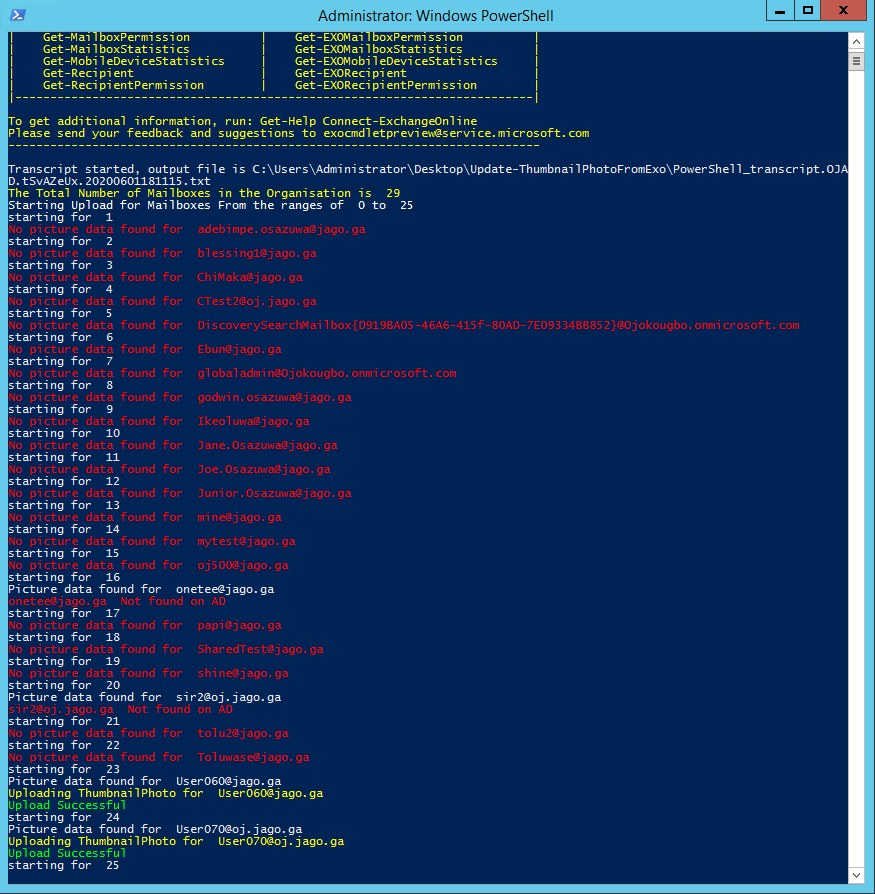
Screenshot of Output from the result of ***UpdateToCloud2.ps1***



Log File from the Above Result



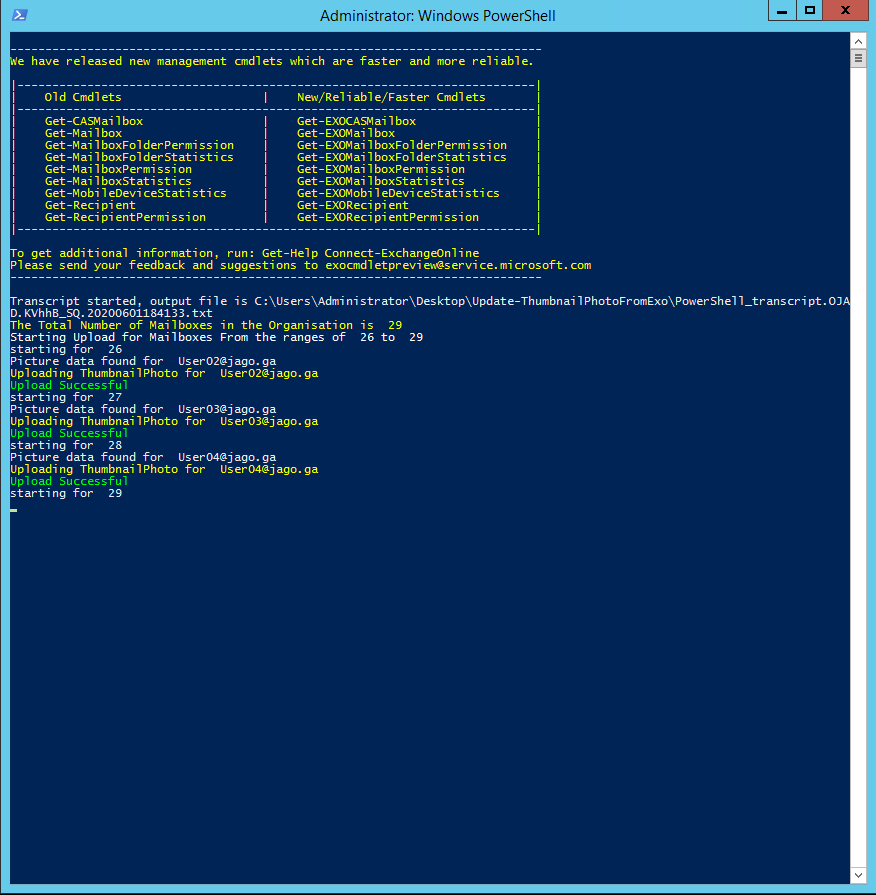
Screenshot of Output from the result of ***UpdateToAD1.ps1***



Log File from the Above Result



Screenshot of Output from the result of ***UpdateToAD2.ps1***



Log File from the Above Result



The below cmdlets can now be used in **TASK SCHEDULER**

UpdateToCloud1.ps1

UpdateToAD1.ps1

UpdateToCloud2.ps1

UpdateToAD2.ps1

**LINK to assist with Task Scheduler:** <https://community.spiceworks.com/how_to/17736-run-powershell-scripts-from-task-scheduler>