# Function New-ExchangeService

## DESCRIPTION

Create new ExchangeService object using Exchange Web Service API

## SYNTAX

New-ExchangeService [-WebServiceUrl <String>] [-WebServiceDll <String>] -UserName <String> -ClearPassword <String> [<CommonParameters>]

New-ExchangeService [-WebServiceUrl <String>] [-WebServiceDll <String>] -UserName <String> -SecurePassword <SecureString> [<CommonParameters>]

## PARAMETERS

### WebServiceUrl

Full Url to Exchange web service. If not specified, Office 365 web service URL is used

### WebServiceDll

Full path to Microsoft.Exchange.WebServices.dll. If not specified, function will try to find it using a default list

### UserName

Exchange user name (e.g "john@contoso.com")

### ClearPassword

Exchange user password, as plain text (not recommended)

### SecurePassword

Exchange user password, as securestring (recommended)

## EXAMPLE

$exchServ = New-ExchangeService -WebServiceUrl https://owa.contoso.com/EWS/Exchange.asmx -WebServiceDll $EWS\_DllPath -UserName "john@contoso.com" -SecurePassword $SecPass

## OUTPUTS

Microsoft.Exchange.WebServices.Data.ExchangeServiceBase type object

# function New-ExchangeMailFolder

## DESCRIPTION

Creates new Exchange mail folder using EWS Managed Api

## SYNTAX

New-ExchangeMailFolder -ExchangeService <ExchangeServiceBase> -NewFolderDisplayName <String> -ParentFolderPath <String> [-WhatIf] [-Confirm] [<CommonParameters>]

New-ExchangeMailFolder -ExchangeService <ExchangeServiceBase> -NewFolderDisplayName <String> -ParentFolderObject <Object> [-WhatIf] [-Confirm] [<CommonParameters>]

## PARAMETERS

### ExchangeService

ExchangeService object. Could be retrieved with function New-ExchangeService

### NewFolderDisplayName

Name of the folder to create

### ParentFolderPath

Full path to parent folder into which to create the new folder. Separate folders with Antislashes ("\"). E.g : "Inbox\Archives\Tests"

### ParentFolderObject

Exchange.WebServices.Data.Folder type object can by specified instead of ParentFolderPath

## EXAMPLE

$newFolderId = New-ExchangeMailFolder -ExchangeService $exchService -NewFolderDisplayName "folder01" -ParentFolderPath "inbox\tests"

## OUTPUTS

Unique Id of the successfully created folder

# Function Get-ExchangeMailFolder

## Description

Returns an object of the Exchange.WebServices.Data.Folder type, for specified path, using Exchange Web Service API

## SYNTAX

Get-ExchangeMailFolder -ExchangeService <ExchangeServiceBase> -FolderId <String> [<CommonParameters>]

Get-ExchangeMailFolder -ExchangeService <ExchangeServiceBase> -FolderPath <String> [-ParentFolder <Object>] [<CommonParameters>]

## Parameters

### ExchangeService

ExchangeService object. Could be retrieved with function New-ExchangeService

### FolderId

Exchange Id of folder

### FolderPath

Full path to folder. Could be supplied instead of folder Exchange Id. Separate folders with Antislashes ("\"). E.g : "Inbox\Archives\Tests"

### ParentFolder

Optional Exchange.WebServices.Data.Folder type object. If supplied, FolderPath will be read starting from this folder instead of Root folder

## EXAMPLE

$archives = Get-ExchangeMailFolder -ExchangeService $ExchangeService -FolderPath "inbox\Archives"

## OUTPUTS

Exchange.WebServices.Data.Folder type object

# Function Get-ExchangeMailSubFolders

## SYNTAX

Get-ExchangeMailSubFolders -ExchangeService <ExchangeServiceBase> -FolderId <String> [<CommonParameters>]

Get-ExchangeMailSubFolders -ExchangeService <ExchangeServiceBase> -FolderPath <String> [<CommonParameters>]

Get-ExchangeMailSubFolders -ExchangeService <ExchangeServiceBase> -FolderObject <Object> [<CommonParameters>]

## DESCRIPTION

Gets all sub-folders for specified Exchange Mail folder using EWS Managed Api

## Parameters

### ExchangeService

ExchangeService object. Could be retrieved with function New-ExchangeService

### FolderId

Exchange Id of folder. Incompatible with FolderPath and FolderObject parameters

### FolderPath

Full path to source folder to move. Incompatible with FolderId and FolderObject parameters. Separate folders with Antislashes ("\"). E.g : "Inbox\Tests"

### FolderObject

Exchange.WebServices.Data.Folder type object. Incompatible with FolderId and FolderPath parameters.

## EXAMPLES

$subfolders = Get-ExchangeMailSubFolders -ExchangeService $exchserv -FolderId "AAMkAGQ5MWNkN2Q3LWE5N..."

$subfolders = Get-ExchangeMailSubFolders -ExchangeService $exchserv -FolderPath "deleted items"

$subfolders = Get-ExchangeMailSubFolders -ExchangeService $exchserv -FolderObject (Get-ExchangeMailFolder -ES $TestExchService -FolderPath inbox)

## OUTPUTS

Array of Microsoft.Exchange.WebServices.Data.Folder objects, or $null if no subfolder found

# function Move-ExchangeMailFolder

## DESCRIPTION

Moves source exchange folder into (under) destination folder.

## SYNTAX

Move-ExchangeMailFolder -ExchangeService <ExchangeServiceBase> -SourceFolderId <String> -DestinationFolderObject <Object> [-WhatIf] [-Confirm] [<CommonParameters>]

Move-ExchangeMailFolder -ExchangeService <ExchangeServiceBase> -SourceFolderId <String> -DestinationFolderPath <String> [-WhatIf] [-Confirm] [<CommonParameters>]

Move-ExchangeMailFolder -ExchangeService <ExchangeServiceBase> -SourceFolderId <String> -DestinationFolderId <String> [-WhatIf] [-Confirm] [<CommonParameters>]

Move-ExchangeMailFolder -ExchangeService <ExchangeServiceBase> -SourceFolderPath <String> -DestinationFolderObject <Object> [-WhatIf] [-Confirm] [<CommonParameters>]

Move-ExchangeMailFolder -ExchangeService <ExchangeServiceBase> -SourceFolderPath <String> -DestinationFolderPath <String> [-WhatIf] [-Confirm] [<CommonParameters>]

Move-ExchangeMailFolder -ExchangeService <ExchangeServiceBase> -SourceFolderPath <String> -DestinationFolderId <String> [-WhatIf] [-Confirm] [<CommonParameters>]

Move-ExchangeMailFolder -ExchangeService <ExchangeServiceBase> -SourceFolderObject <Object> -DestinationFolderObject <Object> [-WhatIf] [-Confirm] [<CommonParameters>]

Move-ExchangeMailFolder -ExchangeService <ExchangeServiceBase> -SourceFolderObject <Object> -DestinationFolderPath <String> [-WhatIf] [-Confirm] [<CommonParameters>]

Move-ExchangeMailFolder -ExchangeService <ExchangeServiceBase> -SourceFolderObject <Object> -DestinationFolderId <String> [-WhatIf] [-Confirm] [<CommonParameters>]

## PARAMETERS

### ExchangeService

ExchangeService object. Could be retrieved with function New-ExchangeService

### SourceFolderId

Exchange Id of folder. Incompatible with SourceFolderPath and SourceFolderObject parameter

### SourceFolderPath

Full path to source folder to move. Incompatible with SourceFolderId and SourceFolderObject parameters. Separate folders with Antislashes ("\"). E.g : "Inbox\Tests"

### SourceFolderObject

Exchange.WebServices.Data.Folder type object. Incompatible with SourceFolderId and SourceFolderPath parameters.

### DestinationFolderId

Exchange Id of folder. Incompatible with DestinationFolderPath and DestinationFolderObject parameters

### DestinationFolderPath

Full path to Destination folder to move. Incompatible with DestinationFolderId and DestinationFolderObject parameters. Separate folders with Antislashes ("\"). E.g : "Inbox\Archives\Tests"

### DestinationFolderObject

Exchange.WebServices.Data.Folder type object. Incompatible with DestinationFolderId and DestinationFolderPath parameters.

## EXAMPLES

Move-ExchangeMailFolder -SourceFolderPath "Inbox\Tests" -DestinationFolderPath "Inbox\Archives"-ExchangeService $exchService

Move-ExchangeMailFolder -SourceFolderObject $SourceFolder -DestinationFolderId "AAMkAGQ5MWNkN2Q3LWE5N..." -ExchangeService $exchService

## OUTPUTS

$True if folder is successfully moved to new location

# Function Rename-ExchangeMailFolder

## DESCRIPTION

Renames Exchange mail folder using EWS Managed Api

## SYNTAX

Rename-ExchangeMailFolder -ExchangeService <ExchangeServiceBase> -FolderNewName <String> -FolderId <String> [-WhatIf] [-Confirm] [<CommonParameters>]

Rename-ExchangeMailFolder -ExchangeService <ExchangeServiceBase> -FolderNewName <String> -FolderPath <String> [-WhatIf] [-Confirm] [<CommonParameters>]

Rename-ExchangeMailFolder -ExchangeService <ExchangeServiceBase> -FolderNewName <String> -FolderObject <Object> [-WhatIf] [-Confirm] [<CommonParameters>]

## Parameters

### ExchangeService

ExchangeService object. Could be retrieved with function New-ExchangeService

### FolderNewName

New name of the folder to update

### FolderId

Exchange Id of folder. Incompatible with FolderPath and FolderObject parameters

### FolderPath

Full path to folder. Separate folders with Antislashes ("\"). E.g : "Inbox\Archives\Tests". Incompatible with FolderId and FolderId parameters

### FolderObject

Exchange.WebServices.Data.Folder type object. Incompatible with FolderPath and FolderId parameters

## EXAMPLE

$Result = Rename-ExchangeMailFolder -ExchangeService $exchService -FolderPath $FolderPath -FolderNewName $NewName

## OUTPUTS

$True if folder is successfully renamed

# Function Remove-ExchangeMailFolder

## DESCRIPTION

Deletes Exchange folder using Exchange Web Services API. Full path (e.g "inbox\archives") or EWS Folder object can be specified.

## SYNTAX

Remove-ExchangeMailFolder -ExchangeService <ExchangeServiceBase> -FolderId <Object> [-DeleteMode <Object>] [-Recurse] [-WhatIf] [-Confirm] [<CommonParameters>]

Remove-ExchangeMailFolder -ExchangeService <ExchangeServiceBase> -FolderPath <Object> [-DeleteMode <Object>] [-Recurse] [-WhatIf] [-Confirm] [<CommonParameters>]

Remove-ExchangeMailFolder -ExchangeService <ExchangeServiceBase> -FolderObject <Object> [-DeleteMode <Object>] [-Recurse] [-WhatIf] [-Confirm] [<CommonParameters>]

## Parameters

### ExchangeService

ExchangeService object. Could be retrieved with function New-ExchangeService

### FolderId

Exchange Id of folder. Incompatible with FolderPath and FolderObject parameter

### FolderPath

Full path to folder. Separate folders with Antislashes ("\"). E.g : "Inbox\Archives\Tests". Incompatible with FolderId and FolderId parameters

### FolderObject

Exchange.WebServices.Data.Folder type object. Incompatible with FolderPath and FolderId parameters

## EXAMPLES

Remove-ExchangeMailFolder -ExchangeService $exs -FolderId "AAMkAGQ5MWNkN2Q3LWE5N..."

Remove-ExchangeMailFolder -ExchangeService $exs -FolderPath "inbox\archives\john" -Hard

Remove-ExchangeMailFolder -ExchangeService $exs -FolderObject $myFolder -Soft

## OUTPUTS

$True if folder(s) removed successfully

## NOTES

The standard and -Hard options are transactional, which means that by the time a web service call completes, the database has moved the item to the Deleted Items folder or permanently removed the item from the Exchange database.

The -Soft delete option works differently for different target versions of Exchange Server. Soft Delete for Exchange 2007 sets a bit on the item that indicates to the Exchange database that the item will be moved to the dumpster folder at an indeterminate time in the future.

Soft Delete for versions of Exchange starting with Exchange 2010, including Exchange Online, immediately moves the item to the dumpster. Soft Delete is not an option for folder deletion. Soft Delete traversal searches for items and folders will not return any results.

On a side note, a folder that previously contained an item (folder,mail) that was not hard deleted (which means, it is still present in the deleted items or dumpster) can only be deleted using the "Hard" delete mode.

# function Clear-ExchangeMailFolder

## DESCRIPTION

Clears a folder using Exchange Web Services Managed Api. Erases all emails within specified folder. Applies to subfolders with -recurse parameter

## SYNTAX

Clear-ExchangeMailFolder -ExchangeService <ExchangeServiceBase> -FolderId <Object> [-DeleteMode <Object>] [-Recurse] [-WhatIf] [-Confirm] [<CommonParameters>]

Clear-ExchangeMailFolder -ExchangeService <ExchangeServiceBase> -FolderPath <Object> [-DeleteMode <Object>] [-Recurse] [-WhatIf] [-Confirm] [<CommonParameters>]

Clear-ExchangeMailFolder-ExchangeService <ExchangeServiceBase> -FolderObject <Object> [-DeleteMode <Object>] [-Recurse] [-WhatIf] [-Confirm] [<CommonParameters>]

## Parameters

### ExchangeService

ExchangeService object. Could be retrieved with function New-ExchangeService

### FolderId

Exchange Id of folder. Incompatible with FolderPath and FolderObject parameters

### FolderPath

Full path to folder. Separate folders with Antislashes ("\"). E.g : "Inbox\Archives\Tests". Incompatible with FolderId and FolderId parameters

### FolderObject

Exchange.WebServices.Data.Folder type object. Incompatible with FolderPath and FolderId parameters

### DeleteMode

Optional. "Default" behaviour is to move all erased emails to the mailbox's Deleted Items folder. "Soft" will move them to the dumpster (items in the dumpster can be recovered). "Hard" will permanently delete the emails.

### Recurse

Optional. Deletes all mails in specified folder + all mails in subfolders

## EXAMPLES

Clear-ExchangeMailFolder -ExchangeService $exchService -FolderPath "inbox\archives\old"

Clear-ExchangeMailFolder -ExchangeService $exchService -FolderPath "inbox\archives" -recurse

Clear-ExchangeMailFolder -ExchangeService $exchService -FolderObject (Get-ExchangeMailFolder -FolderPath "inbox\archives\old")

## OUTPUTS

$True if folder(s) cleared successfully

# function Send-ExchangeMail

## DESCRIPTION

Sends an email using Exchange Web Service Managed Api

## SYNTAX

Send-ExchangeMail -ExchangeService <ExchangeServiceBase> -To <String> -Cc <String> -Bcc <String> -Title <String> [-Body <String>] [-BodyType <String>] [-Attachments <String>] [-Importance <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

Send-ExchangeMail -ExchangeService <ExchangeServiceBase> -To <String> -Bcc <String> -Title <String> [-Body <String>] [-BodyType <String>] [-Attachments <String>] [-Importance <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

Send-ExchangeMail -ExchangeService <ExchangeServiceBase> -To <String> -Cc <String> -Title <String> [-Body <String>] [-BodyType <String>] [-Attachments <String>] [-Importance <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

Send-ExchangeMail -ExchangeService <ExchangeServiceBase> -To <String> -Title <String> [-Body <String>] [-BodyType <String>] [-Attachments <String>] [-Importance <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

Send-ExchangeMail -ExchangeService <ExchangeServiceBase> -Cc <String> -Bcc <String> -Title <String> [-Body <String>] [-BodyType <String>] [-Attachments <String>] [-Importance <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

Send-ExchangeMail -ExchangeService <ExchangeServiceBase> -Cc <String> -Title <String> [-Body <String>] [-BodyType <String>] [-Attachments <String>] [-Importance <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

Send-ExchangeMail -ExchangeService <ExchangeServiceBase> -Bcc <String> -Title <String> [-Body <String>] [-BodyType <String>] [-Attachments <String>] [-Importance <String>] [-WhatIf] [-Confirm] [<CommonParameters>]PARAMETERS

### ExchangeService

ExchangeService object. Could be retrieved with function New-ExchangeService

### To

"Outlook-like" semi-colon separated list of email addresses (optional if Cc or Bcc is specified)

### Cc

"Outlook-like" semi-colon separated list of email addresses (optional if To or Bcc is specified)

### Bcc

"Outlook-like" semi-colon separated list of email addresses (optional if To or Cc is specified)

### Title

"Subject" of the email (mandatory, empty string accepted)

### Body

Mail body (optional, empty string accepted)

### BodyType

"Text" or "HTML". Set body type (optional, Default is HTML).

### Attachments

semi-colon separated list of files full paths

### Importance

(optional) Email priority set by sender, high, low, or normal

## EXAMPLES

$Return = Send-ExchangeMail -ExchangeService $exchserv -To "tom@contoso.com;john@domain.com" -Title 'Hi' -Body $mailbody -BodyType "text"

Send-ExchangeMail -ExchangeService $exchserv -Attachments "c:\docs\file1.txt;c:\pictures\file2.jpg" -Bcc "tom@contoso.com" -Title "Hello" -Body "hello<br><br>world"

## OUTPUTS

$True if mail successfully sent

## NOTES

To, Cc and Bcc are all optional parameters, but one must be specified at least.

# function Send-ExchangeMailReply

## DESCRIPTION

Sends an email reply using Exchange Web Services Managed Api

## SYNTAX

Send-ExchangeMailReply -ExchangeService <ExchangeServiceBase> -ReplyString <String> -MailId <String> [-AddTo <String>] [-AddCc <String>] [-AddBcc <String>] [-Importance <String>] [-Attachments <String>] [-ReplyToAll] [-WhatIf] [-Confirm] [<CommonParameters>]

Send-ExchangeMailReply -ExchangeService <ExchangeServiceBase> -ReplyString <String> -MailObject <Object> [-AddTo <String>] [-AddCc <String>] [-AddBcc <String>] [-Importance <String>] [-Attachments <String>] [-ReplyToAll] [-WhatIf] [-Confirm] [<CommonParameters>]PARAMETERS

### ExchangeService

ExchangeService object. Could be retrieved with function New-ExchangeService

### MailId

Get Email message by its unique Id. Incompatible with MailObject parameter.

### MailObject

Exchange Web Services Email object. Incompatible with MailId parameter. Could be retrieved with function Get-ExchangeMail

### ReplyString

Reply message to be sent ; can be empty, plain text or Html. History of previous email(s) will be kept.

### AddTo

Semi-colon separated list of recipients to add in To recipients

### AddCc

Semi-colon separated list of recipients to add in Cc recipients

### AddBcc

Semi-colon separated list of recipients to add in Bcc recipients

### Attachments

Semi-colon separated list of full path to file(s) to be added as attachment(s)

### ReplyToAll

If specified, all recipients of the initial email will be kept as new recipients. If omitted, only the sender will be kept. Can be associated in both cases with To, Cc, and Bcc parameters.

### Importance

(optional) Email priority set by sender, high, low, or normal

### Forward

(optional) Create 'Forward' response mail instead if Reply mail. At least one recipient must be specified with this option, using 'AddTo' parameter

## EXAMPLES

Send-ExchangeMailReply -ExchangeService $exchService -MailObject $mail -ReplyString "Hello<br><br>World" -Attachments "C:\users\john\doc\prices.pdf;c:\users\john\doc\john.vcf"

Send-ExchangeMailReply -ExchangeService $exchService -MailId "AAMkAGQ5MWNkN2Q3LWE5N..." -ReplyString "Thanks everyone for your cooperation" -AddCc "ceo@contoso.com" -ReplyAll

Send-ExchangeMailReply -es $es -Forward -ReplyString "" -AddTo 'supervision@contoso.com' -MailId "AAMkAGQ5MWNkN2Q3LWE5N..."

## OUTPUTS

$True if mail was successfully sent.

# function Get-ExchangeMail

## DESCRIPTION

Gets Exchange Mail(s) in specified Folder using EWS Managed Api, with several filters on recipients, dates, etc

## SYNTAX

Get-ExchangeMail -ExchangeService <ExchangeServiceBase> -MailId <String> [<CommonParameters>]

Get-ExchangeMail -ExchangeService <ExchangeServiceBase> -FolderId <Object> [-Not] [-SentBefore <String or DateTime>] [-SentAfter <String or DateTime>]  
[-ReceivedBefore <String or DateTime>] [-ReceivedAfter <String or DateTime>] [-From <String>] [-To <String>] [-Cc <String>] [-Bcc <String>] [-DisplayTo   
<String>] [-DisplayCc <String>] [-Subject <String>] [-Body <String>] [-ReadStatus <String>] [-HasAttachments <String>] [-Importance   
<String>] [<CommonParameters>]

Get-ExchangeMail -ExchangeService <ExchangeServiceBase> -FolderPath <Object> [-Not] [-SentBefore <String or DateTime>] [-SentAfter <String or DateTime>]   
[-ReceivedBefore <String or DateTime>] [-ReceivedAfter <String or DateTime>] [-From <String>] [-To <String>] [-Cc <String>] [-Bcc <String>] [-DisplayTo   
<String>] [-DisplayCc <String>] [-Subject <String>] [-Body <String>] [-ReadStatus <String>] [-HasAttachments <String>] [-Importance   
<String>] [<CommonParameters>]

Get-ExchangeMail -ExchangeService <ExchangeServiceBase> -FolderObject <Object> [-Not] [-SentBefore <String or DateTime>] [-SentAfter <String or DateTime>]   
[-ReceivedBefore <String or DateTime>] [-ReceivedAfter <String or DateTime>] [-From <String>] [-To <String>] [-Cc <String>] [-Bcc <String>] [-DisplayTo   
<String>] [-DisplayCc <String>] [-Subject <String>] [-Body <String>] [-ReadStatus <String>] [-HasAttachments <String>] [-Importance   
<String>] [<CommonParameters>]

Get-ExchangeMail -ExchangeService <ExchangeServiceBase> [-Not] [-SentBefore <String or DateTime>] [-SentAfter <String or DateTime>] [-ReceivedBefore  
<String>] [-ReceivedAfter <String or DateTime>] [-From <String>] [-To <String>] [-Cc <String>] [-Bcc <String>] [-DisplayTo <String>] [-DisplayCc   
<String>] [-Subject <String>] [-Body <String>] [-ReadStatus <String>] [-HasAttachments <String>] [-Importance <String>]  
[<CommonParameters>]

## PARAMETERS

### ExchangeService

ExchangeService object. Could be retrieved with function New-ExchangeService

### MailId

Get Email message by its unique Id. Appart from ExchangeService, any other parameter (folder, filters..) becomes irrelevant with this one.

### FolderPath

Full path to Exchange mail folder. Separate folders with Antislashes ("\"). E.g : "Inbox\Archives\Tests"

### FolderObject

Exchange.WebServices.Data.Folder type object can by specified instead of FolderPath

### SentBefore

(optional) Minimum send date (as datetime, or as string, format "yyyy-MM-ddTHH:mm:ss") of the email(s)

### SentAfter

(optional) Maximum send date (as datetime, or as string, format "yyyy-MM-ddTHH:mm:ss") of the email(s)

### ReceivedBefore

(optional) Minimum receive date (as datetime, or as string, format "yyyy-MM-ddTHH:mm:ss") of the email(s)

### ReceivedAfter

(optional) Maximum send date (as datetime, or as string, format "yyyy-MM-ddTHH:mm:ss") of the email(s)

### From

(optional) Exact email address of the Sender

### To

(optional) Matching string in "To" recipients list (can be an full/partial email address or name)

### Cc

(optional) Matching string in "Cc" recipients list (can be an full/partial email address or name)

### Bcc

(optional) Matching string in "Bcc" recipients list (can be an full/partial email address or name)

### DisplayTo

(optional) Matching string in "To" recipients \*names\* list (not addresses)

### DisplayCc

(optional) Matching string in "Cc" recipients \*names\* list (not addresses)

### Subject

(optional, empty string allowed) Matching string in Subject if the email(s)

### Body

(optional, empty string allowed) Matching string in the Body if the email(s) (use with caution for HTML formatted emails)

### ReadStatus

(optional) Status of the email, read or unread

### HasAttachments

(optional) Gets only emails with attachments

### Importance

(optional) Email priority set by sender, high, low, or normal

### Not

(optional) Reverses all other filters, result will exclude all mail that match specified filters. E.g : '-to "john@contoso.com" -subject "alert" -NOT' will exclude all mails sent to John that contain the word "alert" in Subjects

## EXAMPLES

$Mails = Get-ExchangeMail -ExchangeService $exchService -FolderPath "inbox" -Subject "alert" -SentAfter "2021-08-10T23:59:00" -From "network@contoso.com" -Importance "high"

$MailsWithoutAttachments = Get-ExchangeMail -ExchangeService $exchService -FolderPath "inbox\archives" -HasAttachments -Not

## OUTPUTS

Array of Microsoft.Exchange.WebServices.Data.EmailMessage objects, or $null if none found

# function Move-ExchangeMail

## DESCRIPTION

Moves an Exchange Email object to specified folder path or object, using EWS Managed Api. Either Ids, Object, or Path (for destination folder) can be used.

## SYNTAX

Move-ExchangeMail -ExchangeService <ExchangeServiceBase> -MailObject <Object> -DestinationFolderObject <Object> [-WhatIf] [-Confirm] [<CommonParameters>]

Move-ExchangeMail -ExchangeService <ExchangeServiceBase> -MailObject <Object> -DestinationFolderPath <String> [-WhatIf] [-Confirm] [<CommonParameters>]

Move-ExchangeMail -ExchangeService <ExchangeServiceBase> -MailObject <Object> -DestinationFolderId <String> [-WhatIf] [-Confirm] [<CommonParameters>]

Move-ExchangeMail -ExchangeService <ExchangeServiceBase> -MailId <String> -DestinationFolderObject <Object> [-WhatIf] [-Confirm] [<CommonParameters>]

Move-ExchangeMail -ExchangeService <ExchangeServiceBase> -MailId <String> -DestinationFolderPath <String> [-WhatIf] [-Confirm] [<CommonParameters>]

Move-ExchangeMail -ExchangeService <ExchangeServiceBase> -MailId <String> -DestinationFolderId <String> [-WhatIf] [-Confirm] [<CommonParameters>]

## PARAMETERS

### ExchangeService

ExchangeService object. Could be retrieved with function New-ExchangeService

### MailId

Get Email message by its unique Id. Appart from ExchangeService, any other parameter (folder, filters..) becomes irrelevant with this one.

### MailObject

Exchange Web Services Email object. Could be retrieved with function Get-ExchangeMail

### FolderId

Exchange Id of folder.

### DestinationFolderPath

Full path to destination folder. Separate folders with Antislashes ("\"). E.g : "Inbox\Archives\Tests"

### DestinationFolderObject

Exchange.WebServices.Data.Folder type object can by specified instead of DestinationFolderPath

## EXAMPLES

Move-ExchangeMail -ExchangeService $exchService -MailObject $mail -DestinationFolderPath "inbox\archives\folder01"

Move-ExchangeMail -ExchangeService $exchService -MailId "AAMkAGQ5MWNkN2Q3LWE5N..." -DestinationFolderObject $folder

## OUTPUTS

$True if move is successful

# Function Save-ExchangeMailAttachment

## DESCRIPTION

Extracts and saves mail attached files to disk or network share using EWS Managed Api

## SYNTAX

Save-ExchangeMailAttachment -ExchangeService <ExchangeServiceBase> -MailId <String> -DestinationFolder <Object> [-Like <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

Save-ExchangeMailAttachment -ExchangeService <ExchangeServiceBase> -MailObject <Object> -DestinationFolder <Object> [-Like <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

## PARAMETERS

### ExchangeService

ExchangeService object. Could be retrieved with function New-ExchangeService

### MailId

Email message by its unique Id.

### MailObject

Exchange Web Services Email object. Could be retrieved with function Get-ExchangeMail

### DESTINATIONFOLDER

Files will be saved here ; can be either a String with full path to target directory, or System.IO.DirectoryInfo object

### LIKE

(optional) Applied as filter on attached files names

## EXAMPLES

Save-ExchangeMailAttachment -ExchangeService $es -MailObject $MailObj -DestinationFolder "c:\download"

Save-ExchangeMailAttachment -ExchangeService $es -MailId "AAMkAGQ5MWNkN2Q3LWE5N..." -DestinationFolder (Get-Item 'd:\temp') -Like "\*.txt"

## OUTPUTS

$True when all files successfully saved

# function New-ExchangeMeeting

## DESCRIPTION

Creates a new Meeting using Exchange Web Services managed Api

## SYNTAX

New-ExchangeMeeting [-ExchangeService] <ExchangeServiceBase> [-Title] <String> [-Body] <String> [-StartDate] <String> [-EndDate] <String> [[-Location] <String>] [-RequiredAttendees] <String> [[-OptionalAttendees] <String>] [[-Attachments] <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

## Parameters

### ExchangeService

ExchangeService object. Could be retrieved with function New-ExchangeService

### Title

"Subject" of the meeting (empty string allowed)

### Body

Meeting message body (empty string allowed), text-only or HTML

### StartDate

Meeting start date, format yyyy-MM-ddTHH:mm:ss

### EndDate

Meeting end date, format yyyy-MM-ddTHH:mm:ss

### Location

Meeting location (optional, empty string allowed)

### RequiredAttendees

"Outlook-like" semi-colon separated list of email addresses

### OptionalAttendees

"Outlook-like" semi-colon separated list of email addresses

### Attachments

Semi-colon separated list of files full paths (optional)

## EXAMPLES

New-ExchangeMeeting -ExchangeService $es -Title "Presentation" -Body $body -StartDate '2021-08-10T08:30:00' -EndDate '2021-08-10T09:45:00'

New-ExchangeMeeting -ExchangeService $es -Title $title -Body '<p>Hello. This is a test meeting.<br><br>Regards&nbsp;!</p>' -StartDate '2021-08-10T08:30:00' -EndDate '2021-08-10T09:45:00' -Attachments "c:\documents\file1.txt;c:\pictures\image file 2.jpg"

## OUTPUTS

Unique id for created meeting as a String

# function Edit-ExchangeMeeting

## DESCRIPTION

Edit an existing Meeting using Exchange Web Services managed Api

## SYNTAX

Edit-ExchangeMeeting [-ExchangeService] <ExchangeServiceBase> [[-Title] <String>] [[-Body] <String>] [[-StartDate] <String>][[-EndDate] <String>] [[-Location] <String>] [[-RequiredAttendees] <String>] [[-OptionalAttendees] <String>] [[-Attachments] <Array>] [-MeetingId] <String> [-WhatIf] [-Confirm] [<CommonParameters>]

## PARAMETERS

ExchangeService

ExchangeService object. Could be retrieved with function New-ExchangeService

### Title

New "subject" of the meeting (empty string allowed)

### Body

New Meeting message body (empty string allowed), text-only or HTML

### StartDate

New Meeting start date, format yyyy-MM-ddTHH:mm:ss

### EndDate

New Meeting end date, format yyyy-MM-ddTHH:mm:ss

### Location

New Meeting location (empty string allowed)

### RequiredAttendees

New "Outlook-like" semi-colon separated list of email addresses

### OptionalAttendees

New "Outlook-like" semi-colon separated list of email addresses

### Attachments

New semi-colon separated list of files full paths

### MeetingId

Id of the Meeting to modify

## EXAMPLES

Edit-ExchangeMeeting -ExchangeService $es -Body "Modified meeting body message" -MeetingId "BAAAAIIA4AB0xbcQGoLgCAAAAABLH8CjP87WAQAAAAAAAAAAEAAAADA99jEdyitLqpMM8yghhMU="

Edit-ExchangeMeeting -ExchangeService $es -Title "Postponed Meeting" -Body '<p>Hello. This is a modified test meeting.<br><br>Regards&nbsp;!</p>' -StartDate '2021-08-10T08:30:00' -EndDate '2021-08-10T08:45:00' -MeetingId $MeetingId -OptionalAttendees "georges@domain.net"

## OUTPUTS

String with Id of the created meeting, or Null + Exception message if modification failed

# function Stop-ExchangeMeeting

## DESCRIPTION

Deletes or Cancel a Meeting using Exchange Web Services managed Api

## SYNTAX

Stop-ExchangeMeeting [-ExchangeService] <ExchangeServiceBase> [-Delete] [-MeetingId] <String> [-WhatIf] [-Confirm] [<CommonParameters>]

## PARAMETERS

### ExchangeService

ExchangeService object. Could be retrieved with function New-ExchangeService

### Delete

Use "-delete" to completely delete a Meeting

### MeetingId

Id of the Meeting to delete

## EXAMPLES

$MeetingState = Remove-ExchangeMeeting -ExchangeService $es -MeetingId "BAAAAIIA4AB0xbcQGoLgCAAAAAAp1gKGUsnWAQAAAAAAAAAAEAAAAHlrfMPoxtBGv8a7N7md0Zk="

$MeetingState = Remove-ExchangeMeeting -MeetingId $MeetingId -ExchangeService $es -Delete $True

## OUTPUTS

String with Id of the created meeting, or Null + Exception message if creation failed

# function Remove-ExchangeItem

## DESCRIPTION

Removes Exchange Mail or Folder using EWS Managed Api

## SYNTAX

Remove-ExchangeItem [-ExchangeService] <ExchangeServiceBase> [-ExchangeItem] <Object> [[-DeleteMode] <Object>] [-WhatIf] [-Confirm] [<CommonParameters>]

## PARAMETERS

### ExchangeService

ExchangeService object. Could be retrieved with function New-ExchangeService

### ExchangeItem

Exchange Mail or Folder object to delete

### DeleteMode

Optional. "Default" behaviour is to move to the mailbox's Deleted Items folder. "Soft" will move it to the dumpster (items in the dumpster can be recovered). "Hard" will permanently delete the item.

## EXAMPLES

Remove-ExchangeItem -ExchangeService $ExchangeService -ExchangeItem $MailObject

Remove-ExchangeItem -ExchangeService $ExchangeService -ExchangeItem (Get-ExchangeFolder -ExchangeService $exchserv -FolderPath "inbox\archives")

## OUTPUTS

$True if item removed successfully

## NOTES

A folder that previously contained an item (folder, mail) that was not hard deleted (which means, it is still present in the deleted items or dumpster) can only be deleted using the "Hard" delete mode.