Office 365 Delegates Ver 1.0

Contents

Acknowledgement

Thank you to,

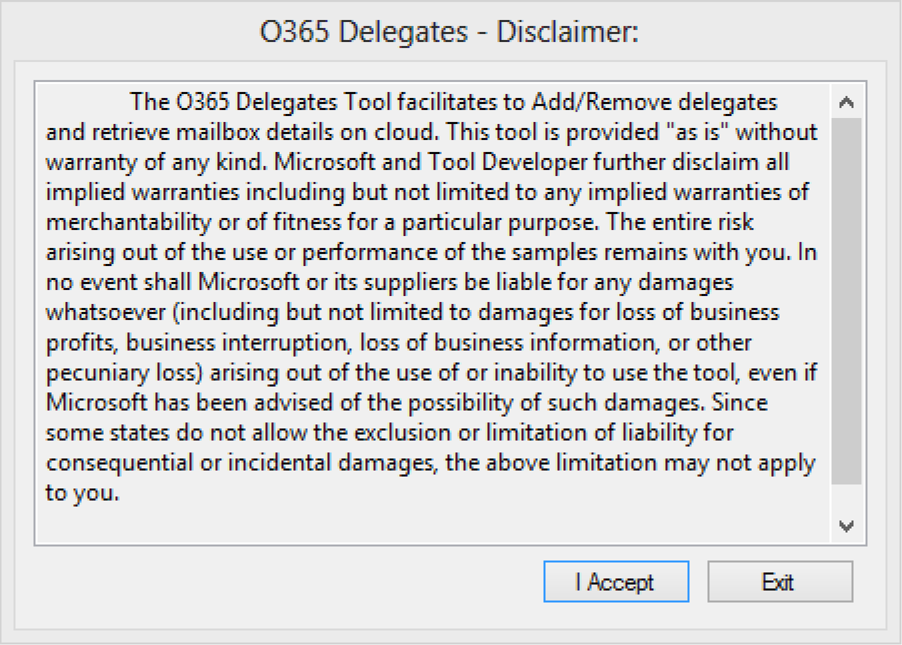
Leena Mathews, Microsoft PSS Support Engineer, for the initial thought.

Venkateswaran Rajagopalan, Microsoft PSS Support Escalation Engineer, for testing the tool.

Go To Contents

Disclaimer

The O365 Delegates Tool facilitates to Add/Remove delegates and retrieve mailbox details on cloud. This tool is provided "as is" without warranty of any kind. Microsoft and Tool Developer further disclaim all implied warranties including but not limited to any implied warranties of merchantability or of fitness for a particular purpose. The entire risk arising out of the use or performance of the samples remains with you. In no event shall Microsoft or its suppliers be liable for any damages whatsoever (including but not limited to damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or inability to use the tool, even if Microsoft has been advised of the possibility of such damages. Since some states do not allow the exclusion or limitation of liability for consequential or incidental damages, the above limitation may not apply to you.



The O365 Delegates Tool facilitates to Add/Remove delegates and retrieve mailbox details on cloud. This tool is provided "as is" without warranty of any kind. Microsoft and Tool Developer further disclaim all implied warranties including but not limited to any implied warranties of merchantability or of fitness for a particular purpose. The entire risk arising out of the use or performance of the samples remains with you. In no event shall Microsoft or its suppliers be liable for any damages whatsoever (including but not limited to damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or inability to use the tool, even if Microsoft has been advised of the possibility of such damages. Since some states do not allow the exclusion or limitation of liability for consequential or incidental damages, the above limitation may not apply to you.

Go To Contents

Introduction

Thank you for choosing the Office 365 Delegates Tool.

This documentation will explain the functionality of the tool, how it works and how it can be used. The Office 365 Delegates Tool has been developed to help Office 365 Exchange administrators to perform get mailbox and the calendar delegate task for the mailboxes in the cloud. Also developers and IT Professional will get benefit out of the sample.

The tool can be used to:

- ✓ Find existing distribution list members.
- ✓ Allow others to manage your mailbox and calendar by giving delegate permission.
- ✓ List delegate permission.
- ✓ Remove delegate permission.
- ✓ Modify delegate permission level.
- ✓ Grant & Revoke "Full Mailbox" and "Send As" Permission.
- ✓ Run various Office 365 PowerShell commands to retrieve user/mailbox details.
- ✓ It is provisioned to get LDP dump of the on-premise user through PowerShell.

As the tool can be used to perform the task that can be done with other application and thru manual PowerShell, but with less effort, it is named as Office 365 Delegates.

Delegates' synonyms: The Person authorized to act as representative for another; a deputy or an agent.

Go To Contents

Pre-Requisites

Operating system: Use Windows 7, Windows 8, Windows Server 2008 R2, or Windows Server 2012.

Microsoft .NET Framework: By Default, the latest OS windows includes .Net Framework. If you do not have, you must install the Microsoft .NET Framework 4.0 feature and above.

Related Link : http://www.microsoft.com/en-us/download/details.aspx?id=17851

Install the Windows Management Framework:

By Default, the latest OS windows includes WinRM and PowerShell. If you do not have, you should download and install the Windows Management Framework. Choose the package that includes

Windows PowerShell v2 and WinRM 2.0, and that applies to your operating system, system architecture, and language.

After you install WinRM and Windows PowerShell, configure the software to work correctly as described in the next steps.

Note If your local computer is protected by a Microsoft Internet Security and Acceleration (ISA) server, you may have to install the Windows Firewall Client or configure a proxy server on your local computer to connect Windows PowerShell to the cloud-based service. For more information, see Windows PowerShell: FAQs for Administrators.

Install Microsoft Online Services Sign-in Assistant: You must install the appropriate version of the Microsoft Online Services Sign-in Assistant for your operating system from the Microsoft Download Center. Microsoft Online Services Sign-In Assistant for IT Professionals RTW.

Install the Windows Azure AD Module for Windows PowerShell: You must install the appropriate version of the Windows Azure AD Module for Windows PowerShell for your operating system from the Microsoft Download Center:

- Windows Azure Active Directory Module for Windows PowerShell (32-bit version)
- Windows Azure Active Directory Module for Windows PowerShell (64-bit version)

Verify that Windows PowerShell can run scripts:

1. Click Start > All Programs > Accessories > Windows PowerShell.
2. Do one of the following to open Windows PowerShell:
   - If you're running Windows Vista, Windows 7, or Windows Server 2008 R2, right-click Windows PowerShell and select Run as administrator. If you get a user account control prompt that asks if you would like to continue, respond Continue.
   - If you're running Windows XP or Windows Server 2003, click Windows PowerShell.
3. Run the following command:

   ```
   Get-ExecutionPolicy
   ```

4. If the value returned is anything other than RemoteSigned, you need to change the value to RemoteSigned.
   Note When you set the script execution policy to RemoteSigned, you can only run scripts that you create on your computer or scripts that are signed by a trusted source.

Enable scripts to run in Windows PowerShell

In Windows PowerShell session you just opened as an administrator, run the following command:

```
Set-ExecutionPolicy RemoteSigned
```

Verify that WinRM allows Windows PowerShell to connect:

1. Click Start > All Programs > Accessories.
2. Do one of the following to open a command prompt:
    - If you're running Windows Vista, Windows 7, or Windows Server 2008 R2, right-click Command Prompt and select Run as administrator. If you get a user account control prompt that asks if you would like to continue, respond Continue.
3. At the command prompt, run the following commands:

    ```
    net start winrm

    winrm get winrm/config/client/auth
    ```

    Note If the WinRM service is already running, you don't have to start it. You can check the status of the WinRM service by running the command sc query winrm.

4. In the results, look for the value Basic = . If the value is Basic = false, you must change the value to Basic = true.
    Note If you started the WinRM service, and you don't need to change the Basic value, run the command net stop winrm to stop the WinRM service.

Configure WinRM to support basic authentication

1. At the command prompt you just opened as an administrator, run the following commands. The value between the braces { } is case-sensitive:

    ```
    winrm set winrm/config/client/auth @{Basic="true"}
    ```

2. In the command output, verify the value Basic = true.
    Note If you started the WinRM service, run the command net stop winrm to stop the WinRM service.

Go To Contents

How Tool Works

As an Office 365 Exchange Administrator, if you want to perform any activity on the mailboxes that are in the cloud, you will use **Remote PowerShell** or **Online Portal**. Both connect to the remote server over the internet.

Similarly, the Office 365 Delegates tool requires internet connection to connect to Office 365 servers.

The tool has to connect to EWS or Remote Powershell to perform the tasks. It uses the **Autodiscover** feature of Exchange to return EWS endpoint to communicate with the Exchange server. By default, direct EWS access is enabled for all Exchange Online plans except for the **Kiosk** plan. So, any mailbox user in the cloud can discover EWS URL using Autodiscover. You do not need any additional permission. However, if you wish to do other tasks in the tool thru EWS, you need to have **Impersonation** permission. When you want to perform a delegate task, the tool makes EWS call to the Exchange Server. So It is crucial to use account with **Impersonation** permission.

You can refer to the following TechNet link for configuring Exchange Impersonation for all users: http://msdn.microsoft.com/en-us/library/bb204095(EXCHG.140).aspx.

Secondly the tool also uses Remote PowerShell to perform various SET & GET tasks. It is expected that the machine that runs the tool installed with pre-requisites for the commands to function. For Example: Get-msoluser command would require additional module to run. So please refer the pre-requisite section in the documentation before using the tool.

Note: I have titled the fields/options as per Exchange Terminology used with Exchange Shell for the easier understanding.

The options in various tabs are un-uniform. It was done intentionally as I wanted to display the different coding style to help IT Pros/Developers.

Note: The EWS tasks being done may take time on the mailboxes on the cloud. Even if the tool completed the task, the replication on the cloud may be still working. When you attempt to perform the activity on the mailbox for which the replication on the cloud, you may get message as "Previous action on the mailbox is still in process. Please try after sometime".

Note: The tool duplicates PowerShell behaviour while executing the commands. That means, if there is no output shown to any of the command executed, you will have no detail shown in output file.

Go To Contents

How to Use the Tool

The tool comes with a simple installation setup.So there is no option to choose the installation path. The setup installs the tool in the following locations and it sub directory:

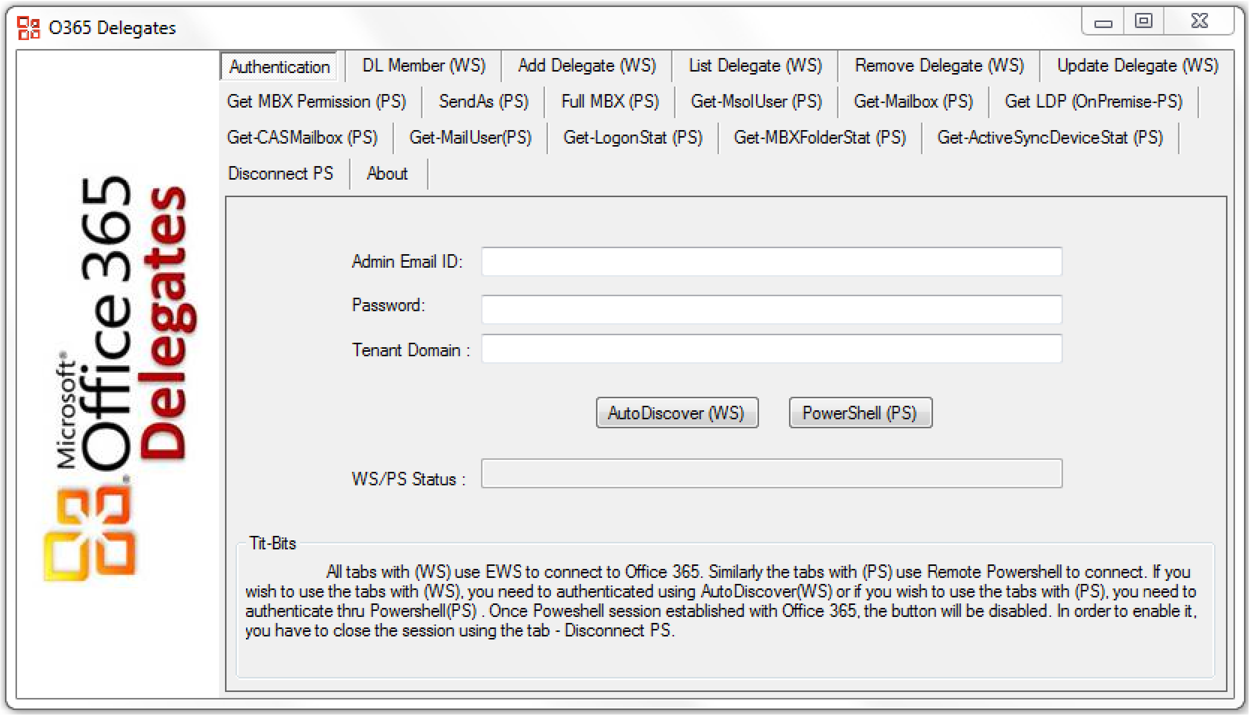
- `C:\Users\<User Name>\AppData\Local\Apps\2.0\....`

We can find the exact installation path after the installation using "Installation Path" option in "About" tab.

It is a stand-alone tool and is multi-instance capable. The tool cannot be run in DOS/Command Prompt as it is GUI based. It creates a Desktop shortcut after installation.

The tool tested in Windows 7, Windows 8, Windows 2008 R2 and Windows 2012.

When you double-click the tool short-cut or the application file, the tool presents you with Disclaimer. If you are fine with the disclaimer, click "I Accept". Then it will display the **Authentication** screen as shown in Picture 1.



Tit-Bits

All tabs with (WS) use EWS to connect to Office 365. Similarly the tabs with (PS) use Remote Powershell to connect. If you wish to use the tabs with (WS), you need to authenticated using AutoDiscover(WS) or if you wish to use the tabs with (PS), you need to authenticate thru Powershell(PS) . Once Poweshell session established with Office 365, the button will be disabled. In order to enable it, you have to close the session using the tab - Disconnect PS.

Picture 1: Authentication screen

The Screen let you input logon details and it is self-explanatory. There are two buttons along with logon input options,

Buttons are,

AutoDiscover (WS): It will use the logon details on the screen to makes autodiscover call to find the Exchange Web Service (EWS) URL and establish EWS session. Once the session established, you can use option in tab titled with (WS). As only EWS session is setup, you will not be able to use the option in tab titled with (PS).

PowerShell (PS): It will use the logon details on the screen to establish PowerShell Session and Import Exchange and MSOnline Modules. Once the session established, you can use option in tab titled with (PS). As only PS session is setup, you will not be able to use the option in tab titled with (WS).

Note: Depending on the internet throughput, the tool takes time to create Remote PowerShell / EWS Session.

In order to discover EWS endpoint, you have to provide the Email ID of any cloud mailbox user with **Impersonation** permission, the password and the tenant domain then click buttons.

You can refer to the following TechNet link for configuring Exchange Impersonation for all users: http://msdn.microsoft.com/en-us/library/bb204095(EXCHG.140).aspx.

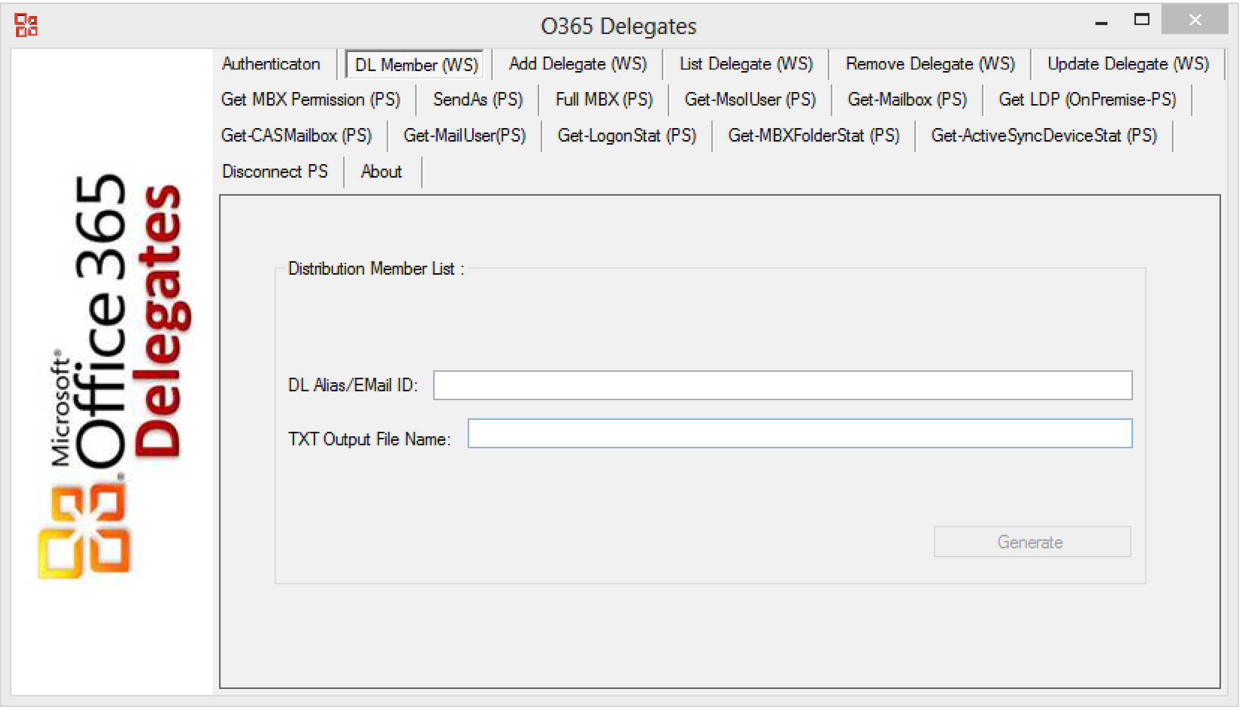
The task buttons in each tab will be enabled only if the EWS endpoints discover process succeeds or the Remote PowerShell Session gets through, otherwise you will not be able to use the tool as EWS Endpoint/Remote PowerShell are crucial to the functionality of the tool.

During the discover/ Remote PowerShell process, the button title changes to “Processing…”. You have to wait until it changes back to its original title. Once the sessions are successfully created, you can use the tool to perform various tasks.

Go To Contents

DL Members:

If you have the Distribution List (not dynamic), and you want to get the members list, you can use this option. Both **DL Alias/Email ID** and **TXT Output File Name** are mandatory.



The value of **DL Alias/Email ID** is verified against Global Address List when you click the **Generate** button. Once all the member details are retrieved from the server, the output is written to the file name given in **TXT Output File Name**.

Please remember that the tool only creates the file. It doesn't create the folder, if the folder path does not exist. In addition to that, the tool cannot create the file in root as the latest Windows versions do not allow the creation without additional permissions.

If only a file name is given without the path, the file gets saved in installation path:

```
C:\Users\<UserName>\AppData\Local\Apps\2.0\..
```

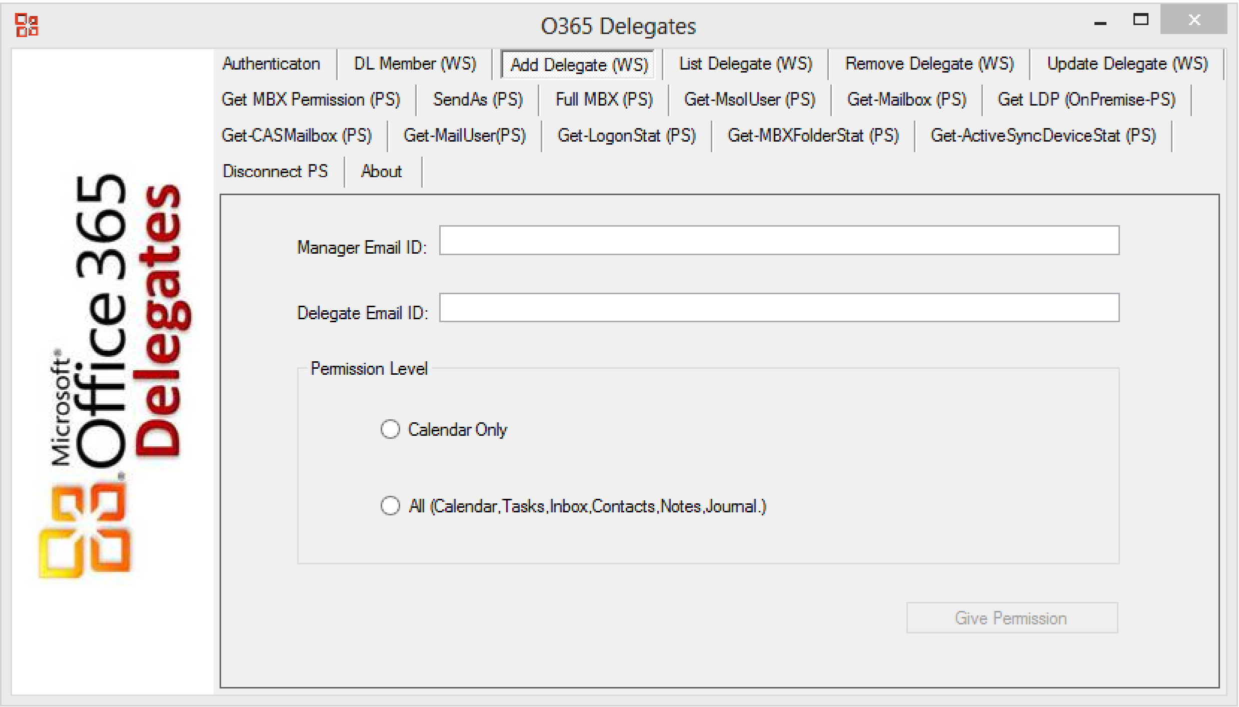
After the member details are written to the file, the file will be opened by the tool and shown on the screen. The recommended file type is TXT.

Go To Contents

Add Delegate:

When Executives want to delegate their assistance to manage their calendar or other folders, the IT Helpdesk will set the required permission on Executive Calendar or other folders using Outlook. This, most of the time, used to be inconvenient as additional resources were involved. The **Add Delegate** option does the same with minimum effort.

All the fields in the tab are mandatory and IDs should be present in the Global Address List. You can give permission to the Delegate on Manager/Executive Mailbox in **Calendar Folder Level** or **All**. When you select **All**, it includes Calendar, Tasks, Inbox, Contacts, Notes and Journal folders. Since most of the time we will give the Editor Permission, there is no option to select the permission to be given on these folders. If you want to change the permission, you can use **Update Delegate** tab.



When "**Give Permission"** button is clicked, the following conditions are checked:

- Managed Email ID and Delegate Email ID are not blank.
- IDs exist in Global Address List.
- The radio button **Calendar Only** or **All (Calendar, Tasks, Inbox, Contacts, Notes, Journal)** is selected.
- If IDs are incomplete and match multiple names in the Global Address List, they are resolved to the first name in the matching list.So give the complete ID.

Go To Contents

List Delegates:

There are situations when you would like to have delegate list of Manager/Executives. **List Delegates** is used to pull the delegate list of Manager/Executives.

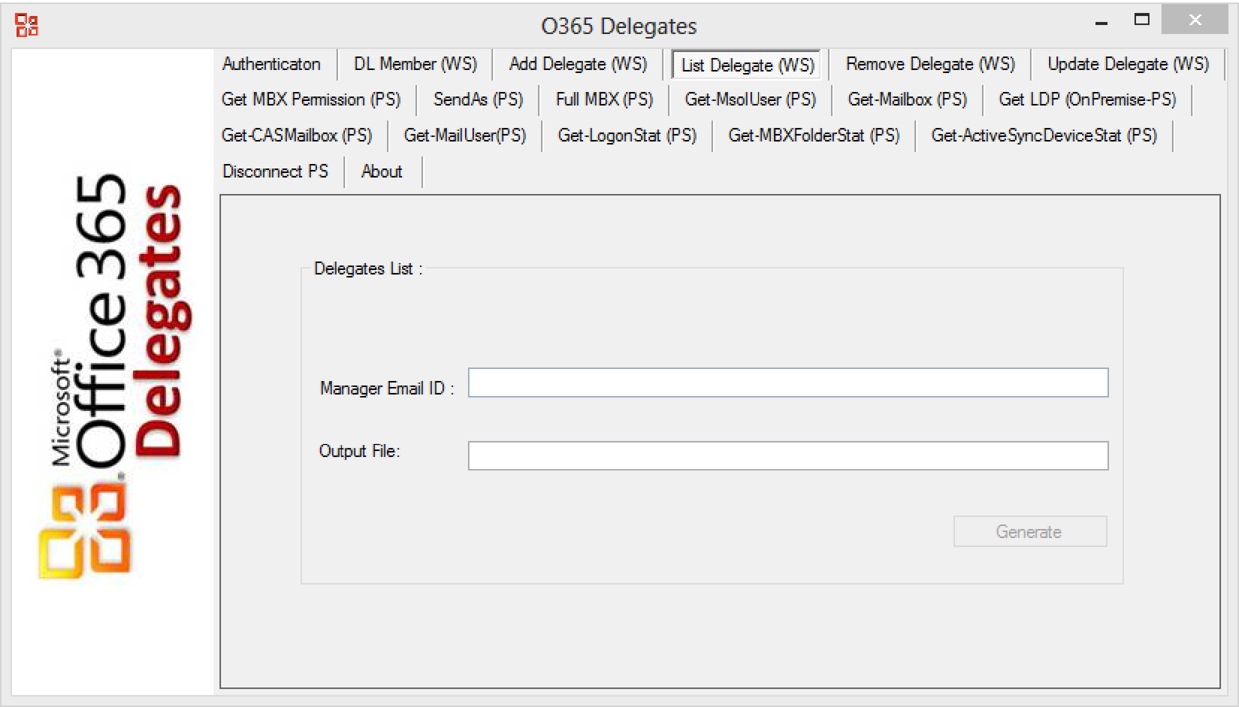
Both the files are mandatory and **Manger Email ID** should exist in Global Address List. If the ID is incomplete and matches multiple names in the Global Address List, it is resolved to the first name in the matching list.

Once all the delegate details are retrieved, the output will be written to the file name given in **Output File**. Please be aware that the tool creates only the file. It will not create the folder if the folder path does not exist. In addition to that, the tool cannot create the file in root as the latest Windows versions do not allow the creation without additional permissions.

If only the file name is given without the path, the file gets saved in:

```
C:\Users\<UserName>\AppData\Local\Apps\2.0\..
```

After delegate details are written to the file, the file will be opened by the tool and displayed on the screen. The recommended file type is TXT.

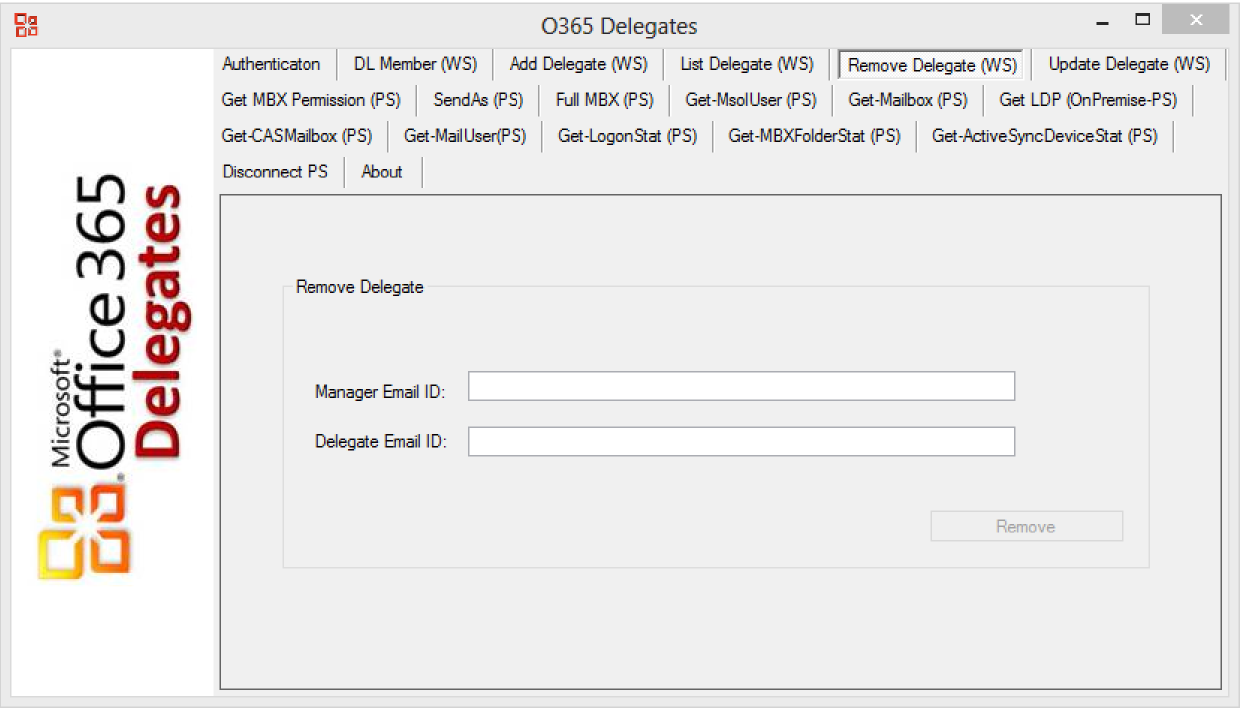


Go To Contents

Remove Delegate:

As the title says, this is the option used to remove the existing delegates from Manager/Executive mailboxes.

Manager and Delegate IDs are mandatory and exist in the Global Address List. If the ID is incomplete and matches multiple names in the Global Address List, it is resolved to the first name in the matching list.



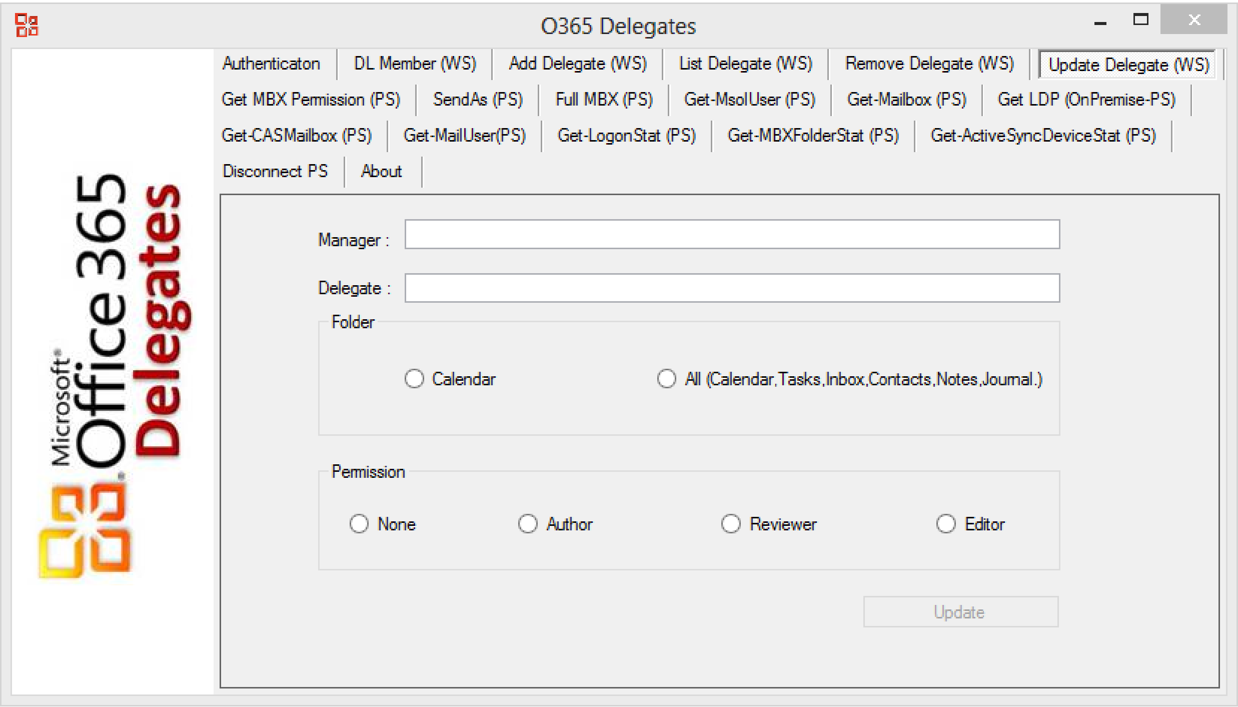
Go To Contents

Update Delegate:

Initially you gave the Editor permission to the delegate, but as the demand changed, you want to change the delegate permission level on the Manager/Executive mailbox.

The option **Update Delegate** is meant for changing the permission of an existing delegate. If the delegate does not have any permission on Manager/Executive mailbox, you cannot use this option.

Manager and Delegate IDs are mandatory and exist in Global Address List. If the ID is incomplete and matches multiple names in the Global Address List, it is resolved to the first name in the matching list.



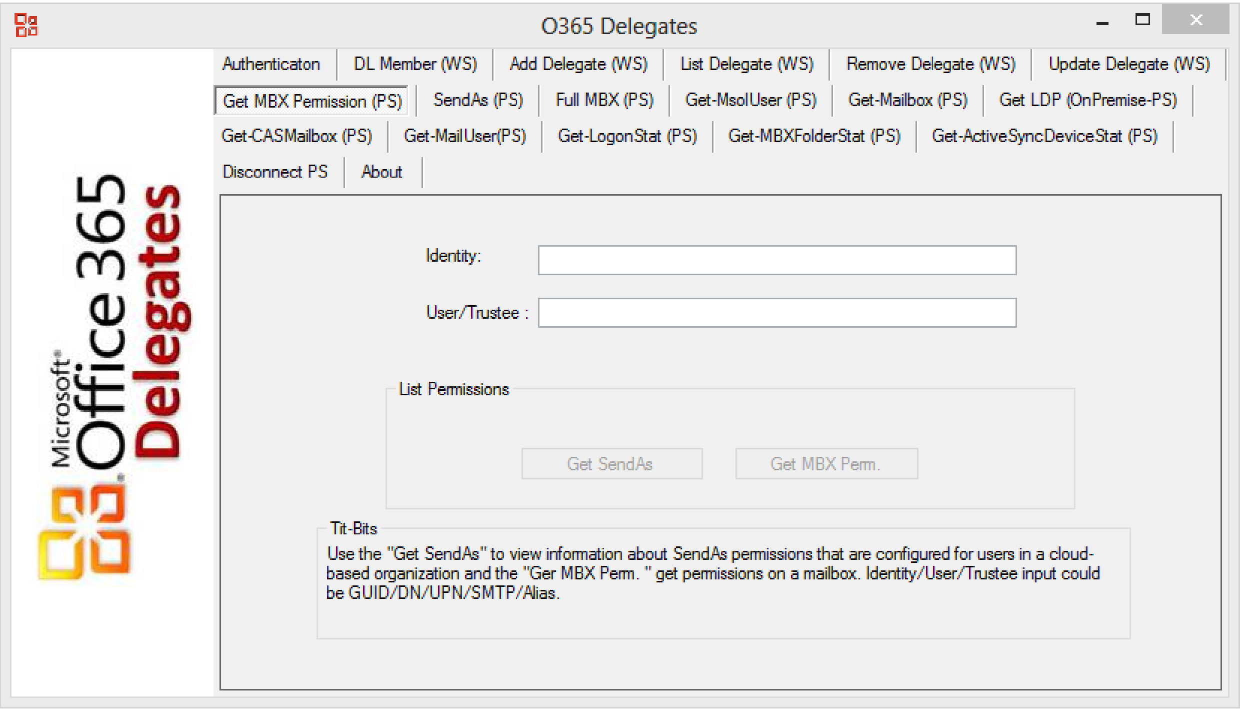
Go To Contents

Get MBX Permission (PS):

With “Get MBX Permission”, you can view “Send As” or “Full Mailbox Permission” configured for the mailbox (Identity).

“User/Trustee” input is optional. If the option is filled, it will retrieve the “Send As” or “Full Mailbox Permission” details available for the “User/Trustee” on the mailbox (Identity).

The tool will show the PowerShell output in a text file and you may save it to your desired location. As I have duplicated the PowerShell output, you may see blank text file, if there was no output to the PowerShell Command.



Identity:

User/Trustee :

List Permissions

Tit-Bits

Use the "Get SendAs" to view information about SendAs permissions that are configured for users in a cloud-based organization and the "Ger MBX Perm. " get permissions on a mailbox. Identity/User/Trustee input could be GUID/DN/UPN/SMTP/Alias.

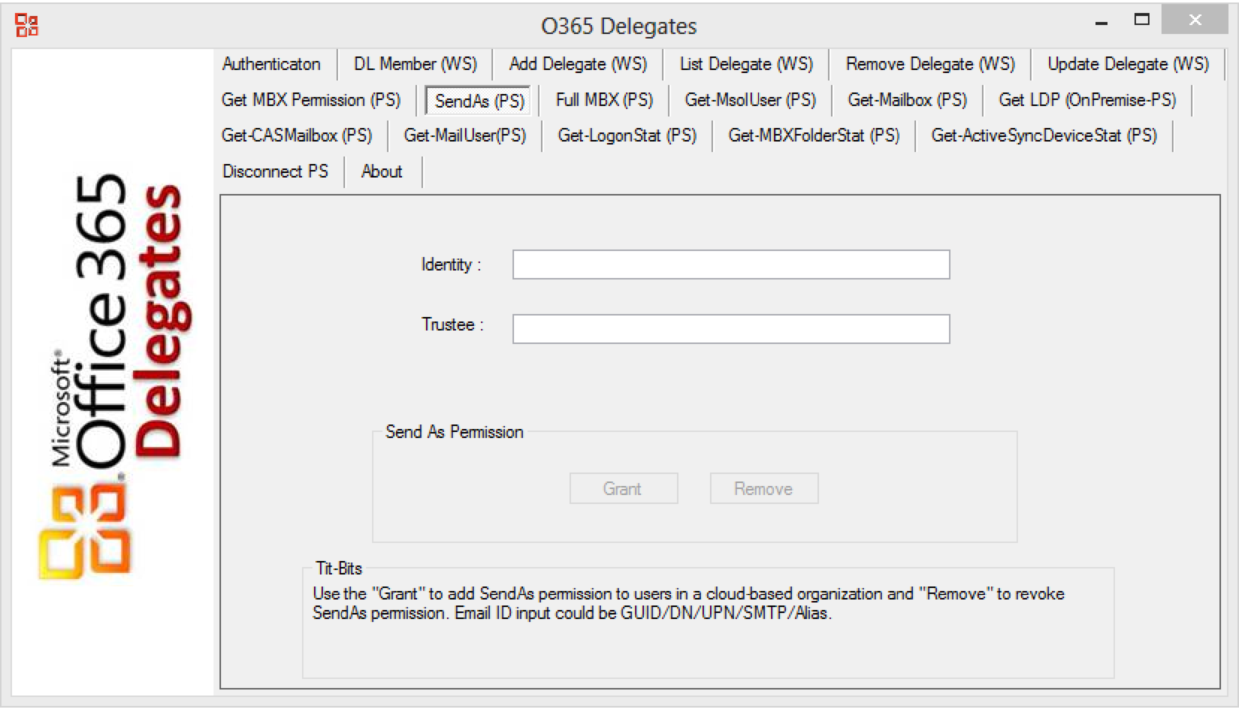
Go To Contents

SendAs (PS):

If you wish to grant or revoke “Send As” permission to a user on a mailbox, you can use “SendAs (PS)”. Both the input fields are required.

Grant/Revoke works on the mailbox (Identity) for the user (Trustee). After the execution, it will list the current permission on the mailbox (Identity).

Input could be in GUID/DN/UPN/SMTP/Alias format.



Use the "Grant" to add SendAs permission to users in a cloud-based organization and "Remove" to revoke SendAs permission. Email ID input could be GUID/DN/UPN/SMTP/Alias.

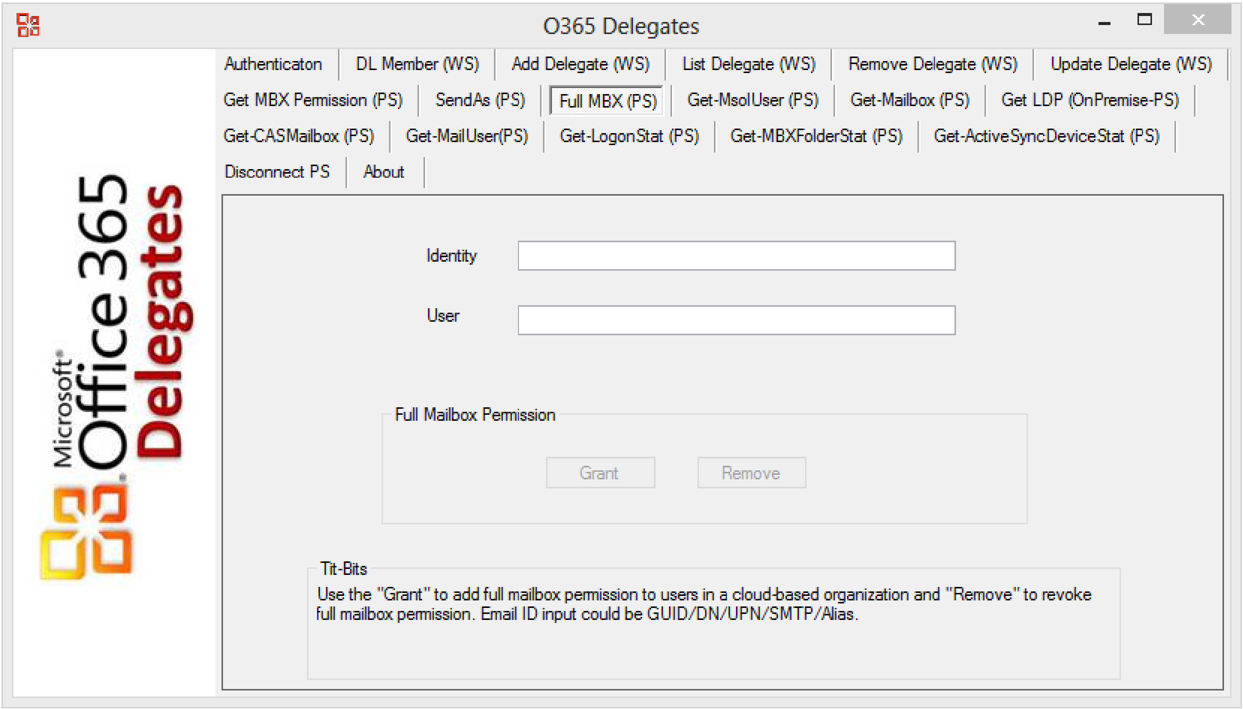
Go To Contents

Full MBX Permission (PS):

The option works in the similar way how the “SendAs (PS)” works. The only difference is, the “Full MBX Permission (PS)” grants/revokes “Full Mailbox” permission to a user on a mailbox. Both the input fields are required.

Grant/Revoke works on the mailbox (Identity) for the user (Trustee). After the execution, it will list the current permission on the mailbox (Identity).

Input could be in GUID/DN/UPN/SMTP/Alias format.



Use the "Grant" to add full mailbox permission to users in a cloud-based organization and "Remove" to revoke full mailbox permission. Email ID input could be GUID/DN/UPN/SMTP/Alias.

[Go To Contents](#)

Get MSOLUser (PS):

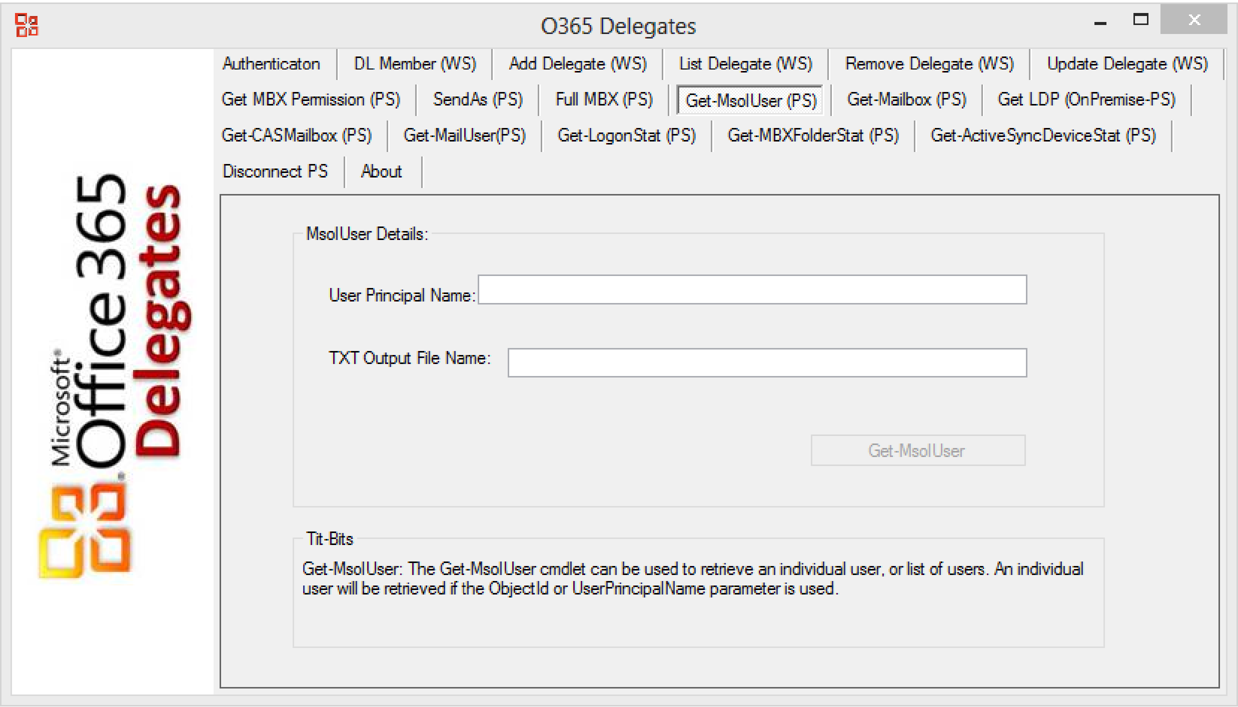
The Get-MsolUser can be used to retrieve an individual user detail on cloud. Both the fields are required.

The output will be written to the file name given in **Output File**. Please be aware that the tool creates only the file. It will not create the folder if the folder path does not exist. In addition to that, the tool cannot create the file in root as the latest Windows versions do not allow the creation without additional permissions.

If only the file name is given without the path, the file gets saved in:

```
C:\Users\<UserName>\AppData\Local\Apps\2.0\..
```

After delegate details are written to the file, the file will be opened by the tool and displayed on the screen. The recommended file type is TXT.



Get-MsolUser: The Get-MsolUser cmdlet can be used to retrieve an individual user, or list of users. An individual user will be retrieved if the ObjectId or UserPrincipalName parameter is used.

Go To Contents

Get-Mailbox (PS):

The Get-Mailbox can be used to retrieve an individual user mailbox detail on cloud. Both the fields are required.

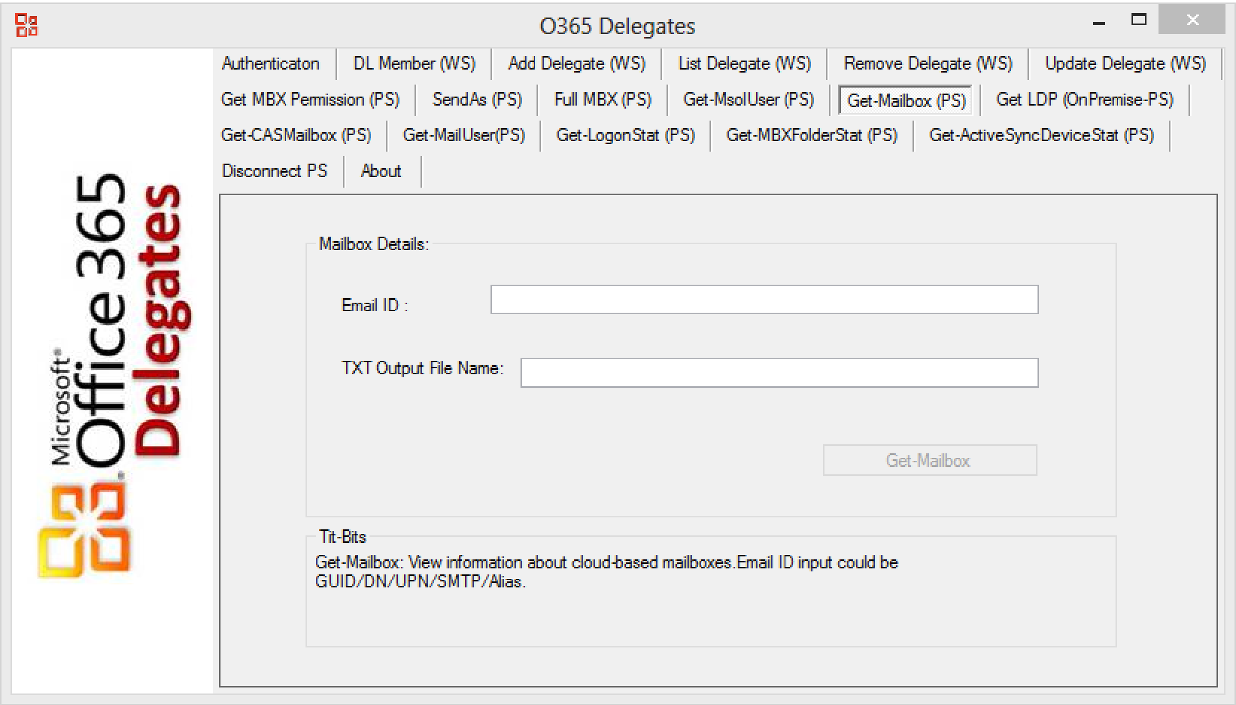
Email ID Input could be in GUID/DN/UPN/SMTP/Alias format.

The output will be written to the file name given in **Output File**. Please be aware that the tool creates only the file. It will not create the folder if the folder path does not exist. In addition to that, the tool cannot create the file in root as the latest Windows versions do not allow the creation without additional permissions.

If only the file name is given without the path, the file gets saved in:

```
C:\Users\<UserName>\AppData\Local\Apps\2.0\..
```

After delegate details are written to the file, the file will be opened by the tool and displayed on the screen. The recommended file type is TXT.



Mailbox Details:

Email ID :

TXT Output File Name:

Tit-Bits

Get-Mailbox: View information about cloud-based mailboxes.Email ID input could be GUID/DN/UPN/SMTP/Alias.

Go To Contents

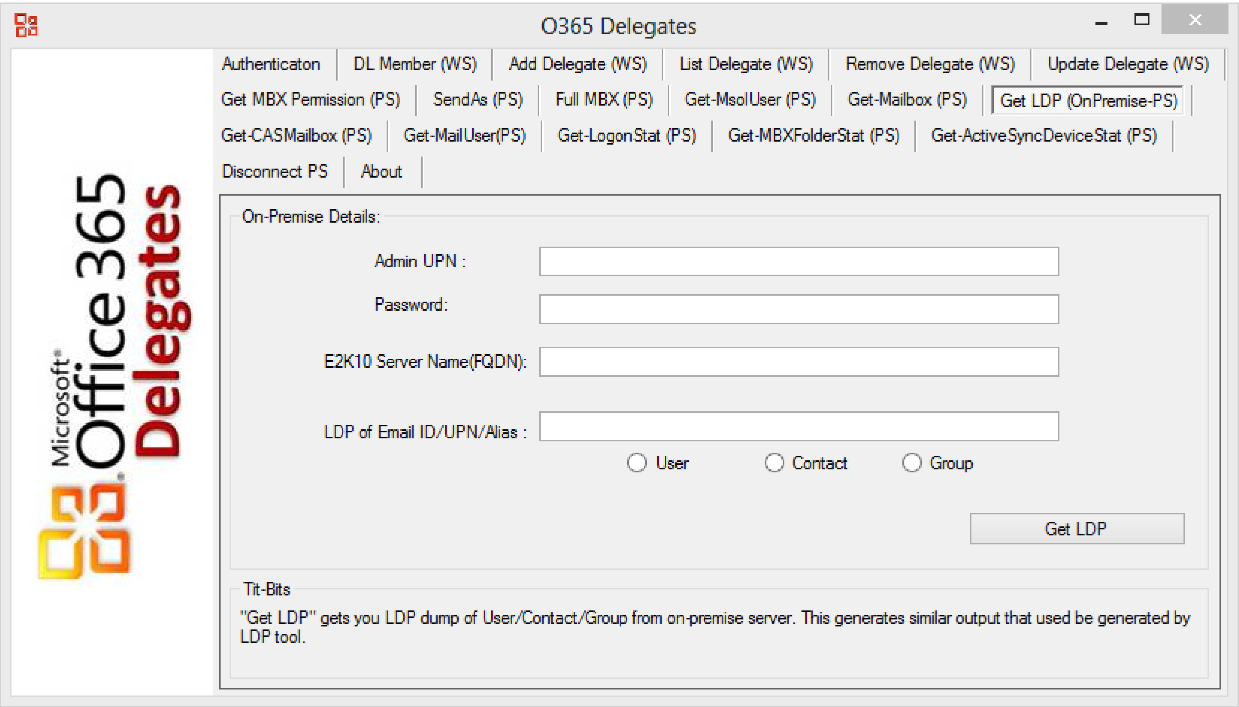
Get LDP (On-Premise PS):

During the exchange troubleshooting most of the support engineers collect LDP dump to narrow down the issue. Collecting LDP require additional tool.

The option “Get LDP (On-Premise PS)” will help you collect the LDP dump of the AD objects – User, Contact and Group. It generates the similar output that used to be generated with LDP tool.

“Get LDP (On-Premise PS)” option is stand alone and it does not require the session creation thru Authentication Tab.

It works with Exchange 2010 and later. The option does not work with object in cloud and works only with On-Premise servers and objects.



"Get LDP" gets you LDP dump of User/Contact/Group from on-premise server. This generates similar output that used be generated by LDP tool.

[Go To Contents](#)

Get-CASMailbox (PS):

The Get-CASMailbox (PS) can be used to retrieve the protocols that are enabled for client connections on cloud. Both the fields are required.

Email ID Input could be in GUID/DN/UPN/SMTP/Alias format.

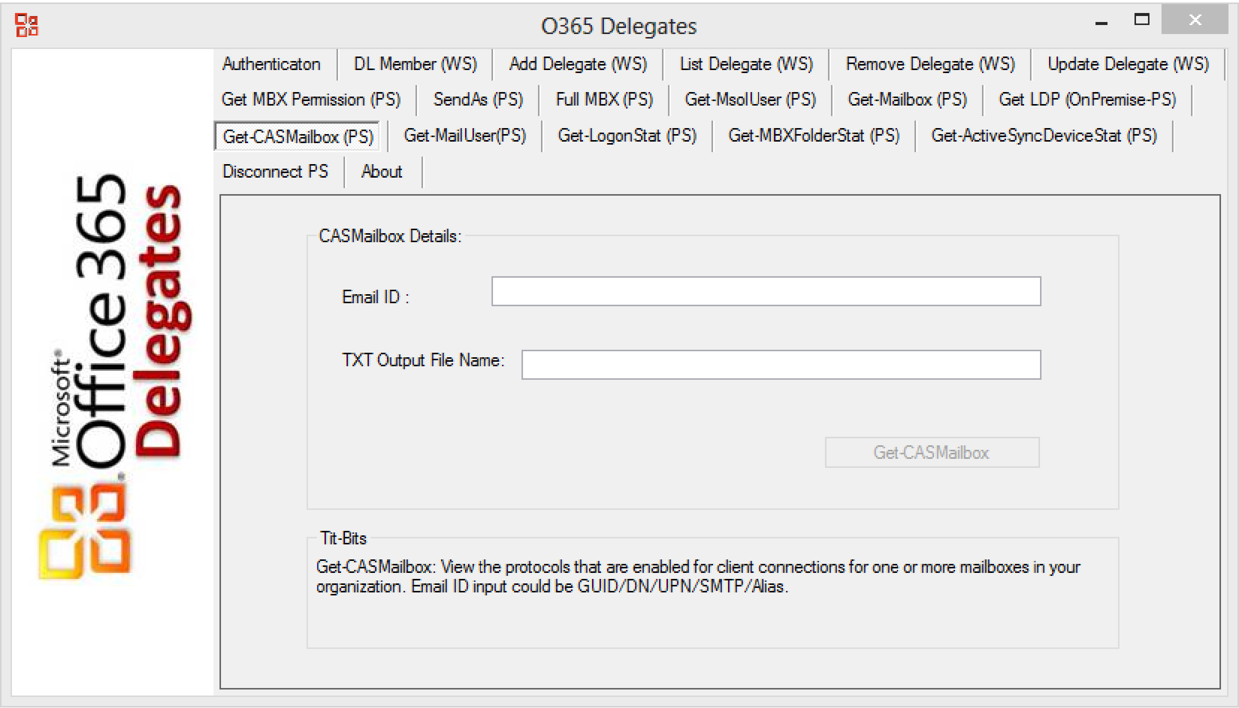
The output will be written to the file name given in **Output File**. Please be aware that the tool creates only the file. It will not create the folder if the folder path does not exist. In addition to that, the tool cannot create the file in root as the latest Windows versions do not allow the creation without additional permissions.

If only the file name is given without the path, the file gets saved in:

```
C:\Users\<UserName>\AppData\Local\Apps\2.0\..
```

After delegate details are written to the file, the file will be opened by the tool and displayed on the screen. The recommended file type is TXT.

Note: As I have duplicated the PowerShell output, you may see blank text file, if the there was no output to the PowerShell Command.



CASMailbox Details:

Email ID :

TXT Output File Name:

Tit-Bits

Get-CASMailbox: View the protocols that are enabled for client connections for one or more mailboxes in your organization. Email ID input could be GUID/DN/UPN/SMTP/Alias.

Go To Contents

Get-MailUser (PS):

The Get-MailUser (PS) can be used to retrieve mail user detail on cloud. Both the fields are required.

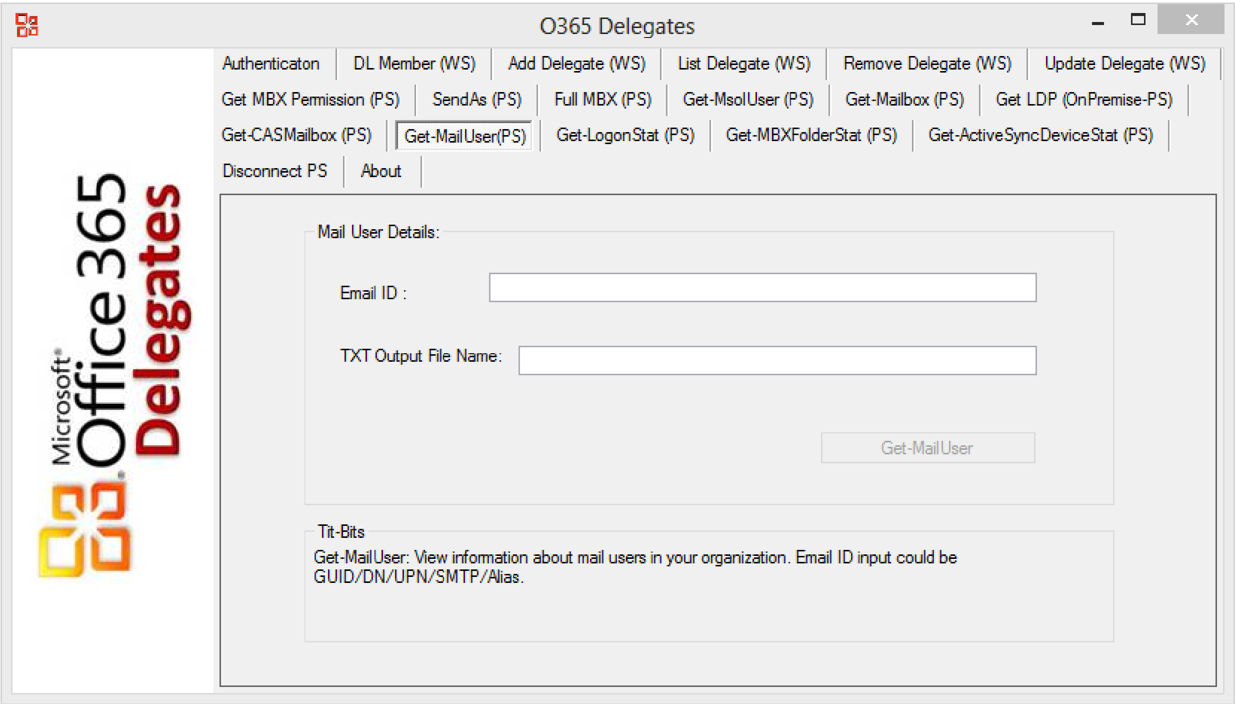
Email ID Input could be in GUID/DN/UPN/SMTP/Alias format.

The output will be written to the file name given in **Output File**. Please be aware that the tool creates only the file. It will not create the folder if the folder path does not exist. In addition to that, the tool cannot create the file in root as the latest Windows versions do not allow the creation without additional permissions.

If only the file name is given without the path, the file gets saved in:

```
C:\Users\<UserName>\AppData\Local\Apps\2.0\..
```

After delegate details are written to the file, the file will be opened by the tool and displayed on the screen. The recommended file type is TXT.



Get-MailUser: View information about mail users in your organization. Email ID input could be GUID/DN/UPN/SMTP/Alias.

[Go To Contents](#)

Get-LogonStat (PS):

The Get-LogonStat (PS) can be used to retrieve information about open logon sessions to a specified mailbox, such as user name, logon time, and last access time. A user must sign out to close a logon session; therefore multiple sessions may appear for users who just close their browser.

Email ID Input could be in GUID/DN/UPN/SMTP/Alias format. Both the fields are required.

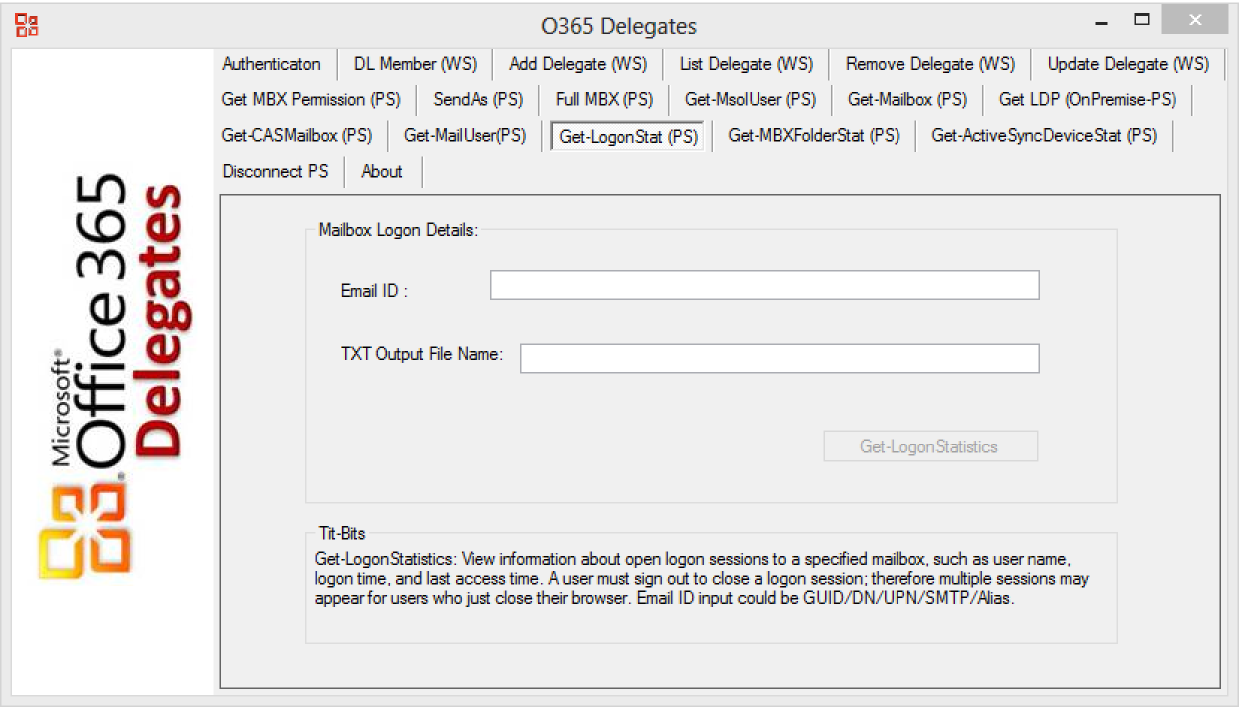
The output will be written to the file name given in **Output File**. Please be aware that the tool creates only the file. It will not create the folder if the folder path does not exist. In addition to that, the tool cannot create the file in root as the latest Windows versions do not allow the creation without additional permissions.

If only the file name is given without the path, the file gets saved in:

```
C:\Users\<UserName>\AppData\Local\Apps\2.0\..
```

After delegate details are written to the file, the file will be opened by the tool and displayed on the screen. The recommended file type is TXT.

Note: As I have duplicated the PowerShell output, you may see blank text file, if the there is no output to the PowerShell Command.



Get-LogonStatistics: View information about open logon sessions to a specified mailbox, such as user name, logon time, and last access time. A user must sign out to close a logon session; therefore multiple sessions may appear for users who just close their browser. Email ID input could be GUID/DN/UPN/SMTP/Alias.

Go To Contents

Get-MBXFolderStat (PS):

The Get-MBXFolderStat (PS) can be used to retrieve information about the folders in a specified mailbox, including the number and size of items in the folder, the folder name and ID, and other information

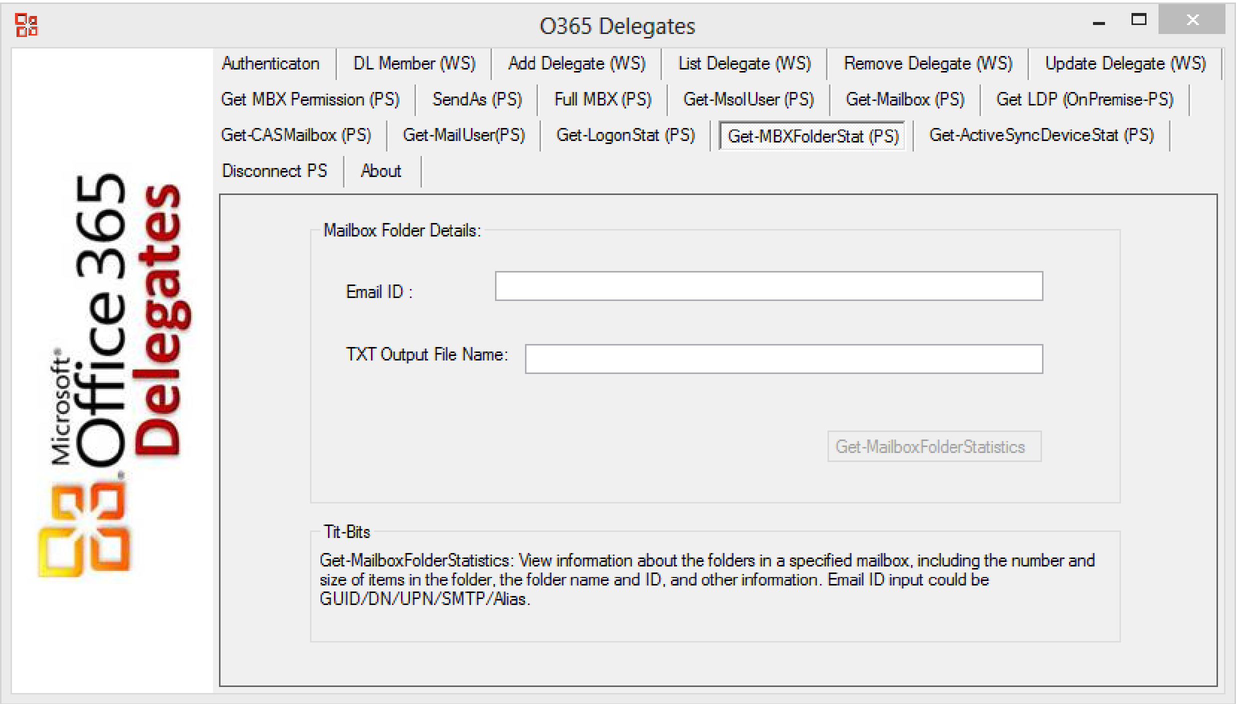
Email ID Input could be in GUID/DN/UPN/SMTP/Alias format. Both the fields are required.

The output will be written to the file name given in **Output File**. Please be aware that the tool creates only the file. It will not create the folder if the folder path does not exist. In addition to that, the tool cannot create the file in root as the latest Windows versions do not allow the creation without additional permissions.

If only the file name is given without the path, the file gets saved in:

```
C:\Users\<UserName>\AppData\Local\Apps\2.0\..
```

After delegate details are written to the file, the file will be opened by the tool and displayed on the screen. The recommended file type is TXT.



Mailbox Folder Details:

Email ID :

TXT Output File Name:

Tit-Bits

Get-MailboxFolderStatistics: View information about the folders in a specified mailbox, including the number and size of items in the folder, the folder name and ID, and other information. Email ID input could be GUID/DN/UPN/SMTP/Alias.

Get-ActiveSyncDeviceStat (PS):

The Get-ActiveSyncDeviceStat (PS) can be used to retrieve the list of mobile phones configured to synchronize with a specified user's mailbox and synchronization statistics for each device. The information returned includes statistics such as the last time synchronization was attempted and the device identification

Email ID Input could be in GUID/DN/UPN/SMTP/Alias format. Both the fields are required.

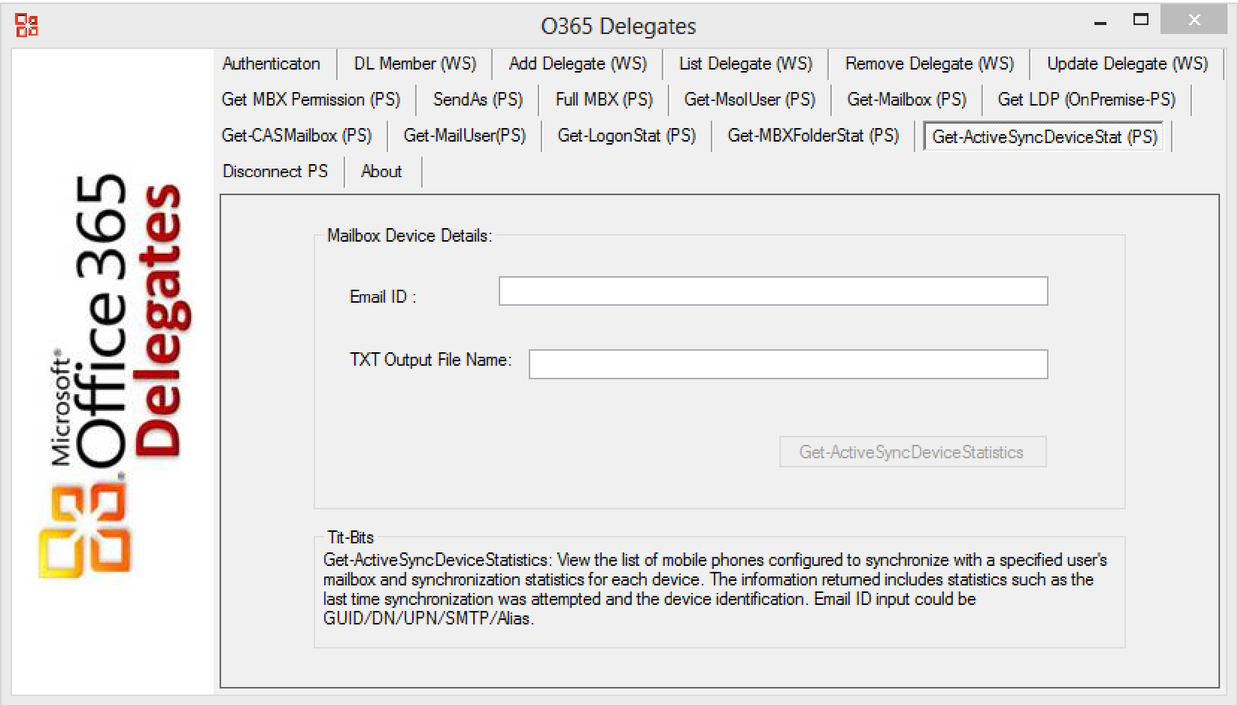
The output will be written to the file name given in **Output File**. Please be aware that the tool creates only the file. It will not create the folder if the folder path does not exist. In addition to that, the tool cannot create the file in root as the latest Windows versions do not allow the creation without additional permissions.

If only the file name is given without the path, the file gets saved in:

```
C:\Users\<UserName>\AppData\Local\Apps\2.0\..
```

After delegate details are written to the file, the file will be opened by the tool and displayed on the screen. The recommended file type is TXT.

Note: As I have duplicated the PowerShell output, you may see blank text file, if the there is no output to the PowerShell Command.



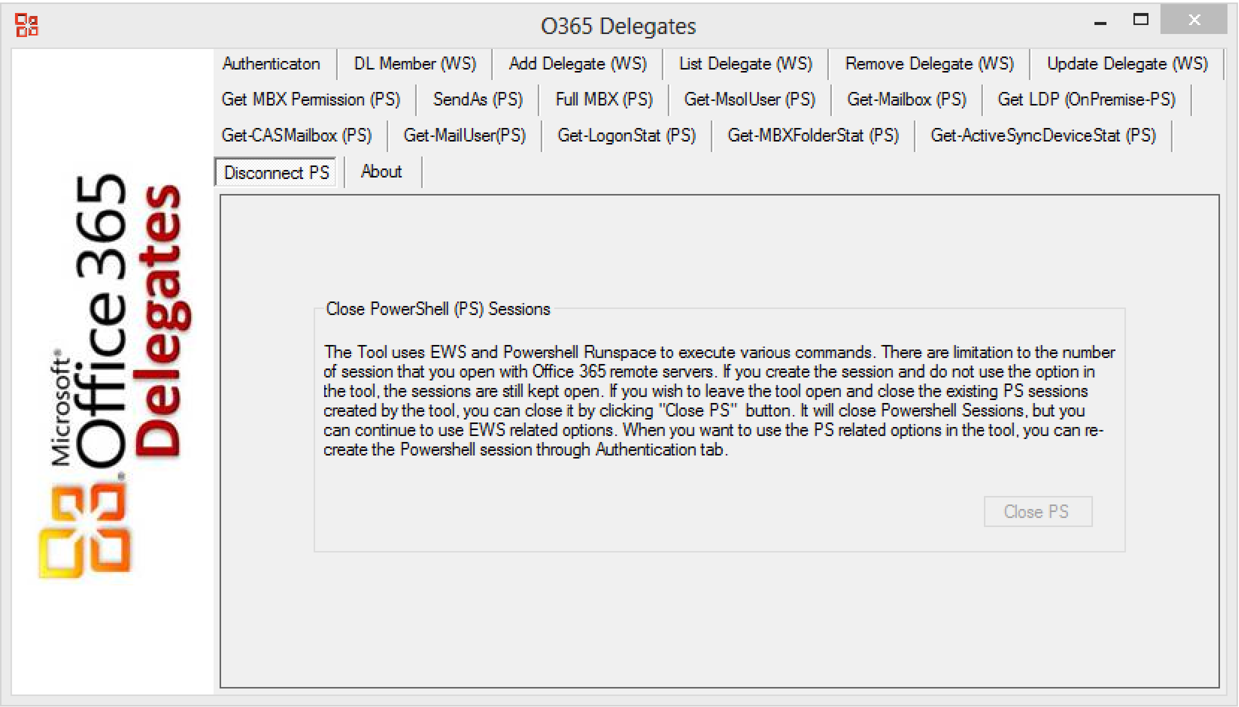
Get-ActiveSyncDeviceStatistics: View the list of mobile phones configured to synchronize with a specified user's mailbox and synchronization statistics for each device. The information returned includes statistics such as the last time synchronization was attempted and the device identification. Email ID input could be GUID/DN/UPN/SMTP/Alias.

Go To Contents

Disconnect PS:

The Tool uses EWS and Powershell Runspace to execute various commands. There are limitation to the number of session that you open with Office 365 remote servers. If you create the session and do not use the option in the tool, the sessions are still kept open. If you wish to leave the tool open and close the existing PS sessions created by the tool, you can close it by clicking "Close PS" button. It will close Powershell Sessions, but you can continue to use EWS related options. When you want to use the PS related options in the tool, you can re-create the Powershell session through Authentication tab.

The button “Close PS” available to use, only after successful PowerShell Session creation through Authentication Tab.



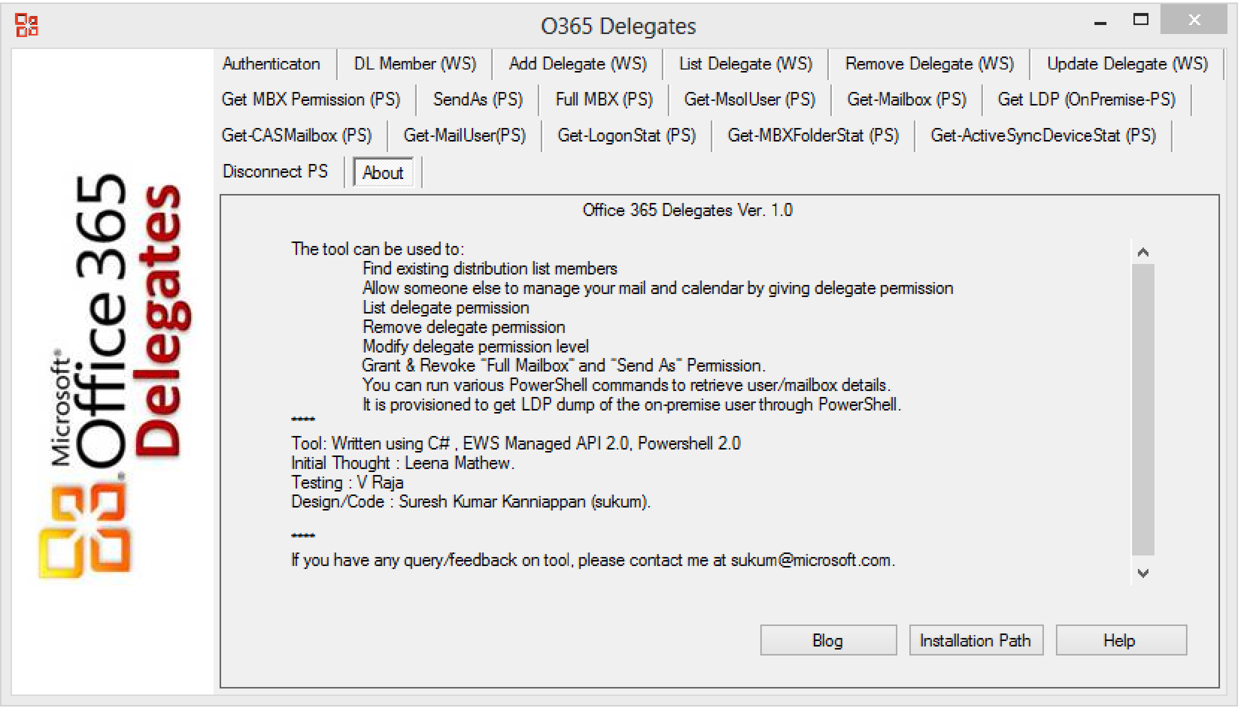
The Tool uses EWS and Powershell Runspace to execute various commands. There are limitation to the number of session that you open with Office 365 remote servers. If you create the session and do not use the option in the tool, the sessions are still kept open. If you wish to leave the tool open and close the existing PS sessions created by the tool, you can close it by clicking "Close PS" button. It will close Powershell Sessions, but you can continue to use EWS related options. When you want to use the PS related options in the tool, you can re-create the Powershell session through Authentication tab.

Go To Contents

About:

As the tool is getting installed using simple instaltion kit, there is no control to select location of the instllation path. After installation, if you wish to know the installation path, you can find it thru button “Installation Path”.

You may also reach to the developers of the tool thru blog. The button “Blog” will take you to blog internet site.



Office 365 Delegates Ver. 1.0

The tool can be used to:
Find existing distribution list members
Allow someone else to manage your mail and calendar by giving delegate permission
List delegate permission
Remove delegate permission
Modify delegate permission level
Grant & Revoke "Full Mailbox" and "Send As" Permission.
You can run various PowerShell commands to retrieve user/mailbox details.
It is provisioned to get LDP dump of the on-premise user through PowerShell.
****
Tool: Written using C# , EWS Managed API 2.0, Powershell 2.0
Initial Thought : Leena Mathew.
Testing : V Raja
Design/Code : Suresh Kumar Kanniappan (sukum).

****
If you have any query/feedback on tool, please contact me at sukum@microsoft.com.

Go To Contents

Known Issues

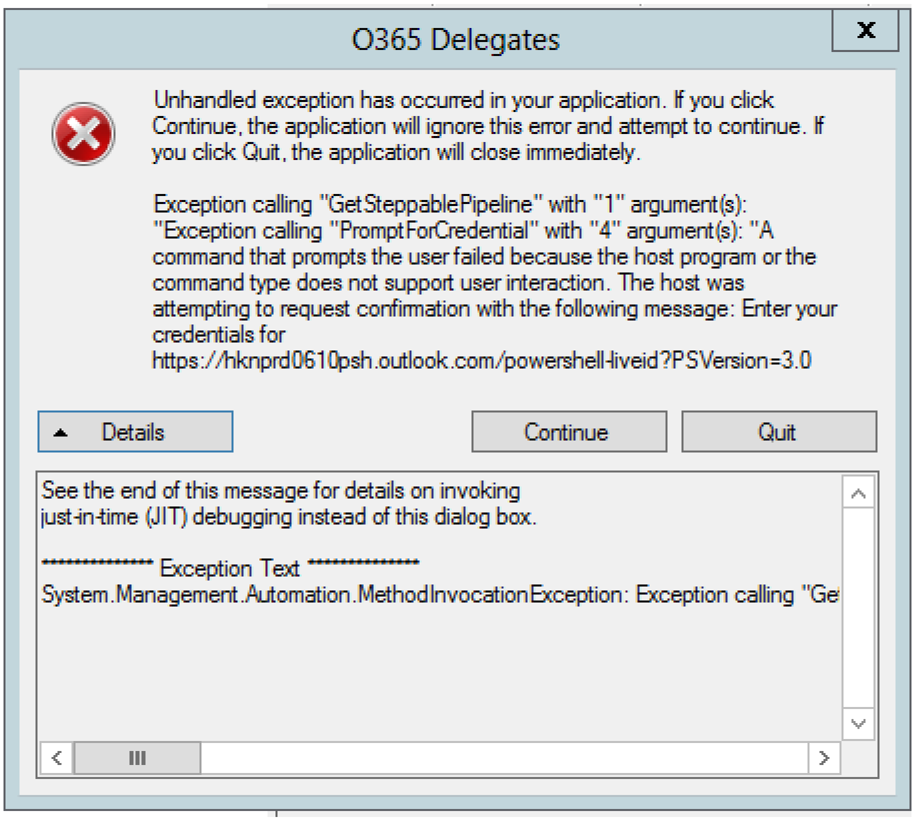
1. "Get LDP (On-Premise PS)" will return some of the value as "System._ComObject".
2. If you leave the Remote PowerShell session running for a long time without using the tool, then you try to use the PS related option, if you may get unexpected behaviour.

Go To Contents

Troubleshooting

Issue #1:

PowerShell Session to Office 365 will expire if it is not actively used. If you create the session and do not use the tool for a long time, then try to use it after session expiry, you will see the below error. You have to reopen the tool and create new PowerShell Session thru Authentication Tab.



Unhandled exception has occurred in your application. If you click Continue, the application will ignore this error and attempt to continue. If you click Quit, the application will close immediately.

Exception calling "GetSteppablePipeline" with "1" argument(s): "Exception calling "PromptForCredential" with "4" argument(s): "A command that prompts the user failed because the host program or the command type does not support user interaction. The host was attempting to request confirmation with the following message: Enter your credentials for https://hknprd0610psh.outlook.com/powershell-liveid?PSVersion=3.0

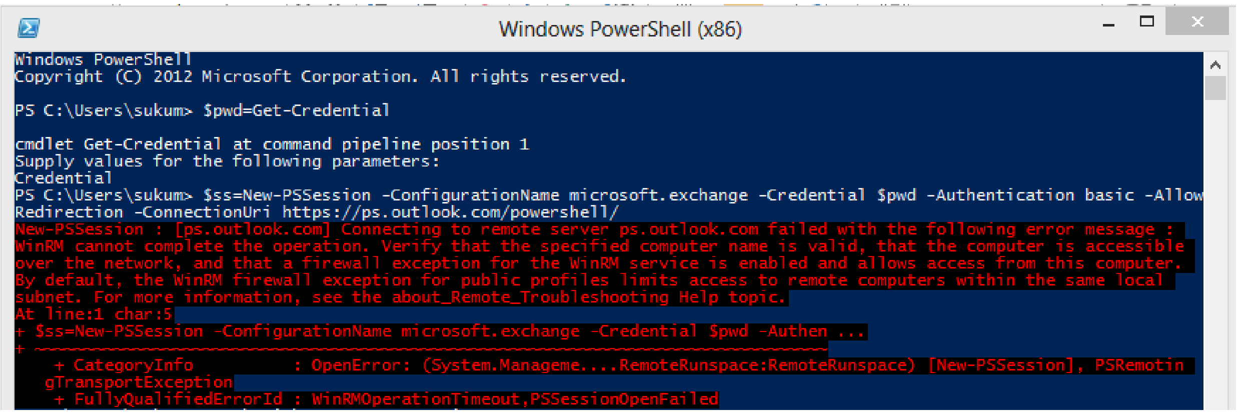
```
See the end of this message for details on invoking
just-in-time (JIT) debugging instead of this dialog box.

************** Exception Text **************
System.Management.Automation.MethodInvocationException: Exception calling "Ge
```

Troubleshooting: Restart the tool , the create new session and check.

Issue #2:

If you are not able to create Remote PowerShell Session to cloud. Please check whether you are able to connect to Remote PowerShell manually following the method detail in the article : http://help.outlook.com/en-us/140/cc952755.aspx

If you receive error like,

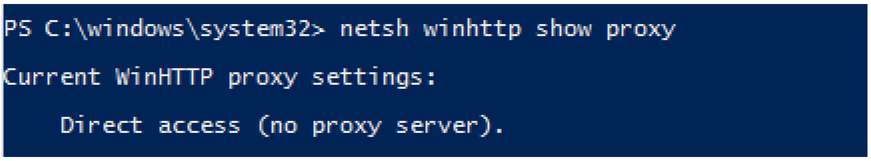


```
Windows PowerShell
Copyright (C) 2012 Microsoft Corporation. All rights reserved.

PS C:\Users\sukum> $pwd=Get-Credential

cmdlet Get-Credential at command pipeline position 1
Supply values for the following parameters:
Credential
PS C:\Users\sukum> $ss=New-PSSession -ConfigurationName microsoft.exchange -Credential $pwd -Authentication basic -Allow
Redirection -ConnectionUri https://ps.outlook.com/powershell/
New-PSSession : [ps.outlook.com] Connecting to remote server ps.outlook.com failed with the following error message :
WinRM cannot complete the operation. Verify that the specified computer name is valid, that the computer is accessible
over the network, and that a firewall exception for the WinRM service is enabled and allows access from this computer.
By default, the WinRM firewall exception for public profiles limits access to remote computers within the same local
subnet. For more information, see the about_Remote_Troubleshooting Help topic.
At line:1 char:5
+ $ss=New-PSSession -ConfigurationName microsoft.exchange -Credential $pwd -Authen ...
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : OpenError: (System.Manageme....RemoteRunspace:RemoteRunspace) [New-PSSession], PSRemotin
   gTransportException
    + FullyQualifiedErrorId : WinRMOperationTimeout,PSSessionOpenFailed
```

Please set the proper proxy server.

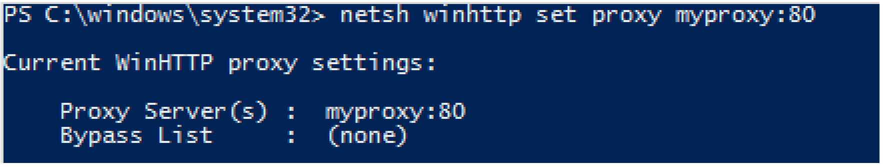
Troubleshooting:

Run the command: netsh winhttp show proxy and check what the proxy server is set. If you are using proxy, but the output of the command shown as below, set the correct proxy server.



```
PS C:\windows\system32> netsh winhttp show proxy

Current WinHTTP proxy settings:

    Direct access (no proxy server).
```

To set the proxy server , run the command : netsh winhttp set proxy <Proxy Server Name:Port>



```
PS C:\windows\system32> netsh winhttp set proxy myproxy:80

Current WinHTTP proxy settings:

    Proxy Server(s) :  myproxy:80
    Bypass List     :  (none)
```

If you are not able to connect to PowerShell by following the method in the link : http://help.outlook.com/en-us/140/cc952755.aspx, the PS related options in the tool will not work.

Issue #3:

At times, all would look good but the session may not be established due to server connection rejection or intermittent issue or transient failure.

Troubleshooting: Close the tool and try to create the session after a couple of minutes.

Issue #4:

There are situation, you could successfully create the session and run PowerShell command, but the MSOnline Service related command like get-MsolUser may not work as the module did not get imported or did not get connected. The PowerShell command will not be recognised at all.

Troubleshooting: Check tool Pre-Requisite are installed. If the below modules are not in place, your commands may not be recognised.

Install Microsoft Online Services Sign-in Assistant: You must install the appropriate version of the Microsoft Online Services Sign-in Assistant for your operating system from the Microsoft Download Center. Microsoft Online Services Sign-In Assistant for IT Professionals RTW.

Install the Windows Azure AD Module for Windows PowerShell: You must install the appropriate version of the Windows Azure AD Module for Windows PowerShell for your operating system from the Microsoft Download Center:

- Windows Azure Active Directory Module for Windows PowerShell (32-bit version)
- Windows Azure Active Directory Module for Windows PowerShell (64-bit version)

Issue #5:

There are situation, you could successfully create the session and run PowerShell command, but the MSOnline Service related command like get-MsolUser may not work as the module did not get imported or did not get connected. The command will not be recognised at all.

Troubleshooting: Close the PS Session thru "Close PS" option in "Disconnect PS" tab and try to create the session.

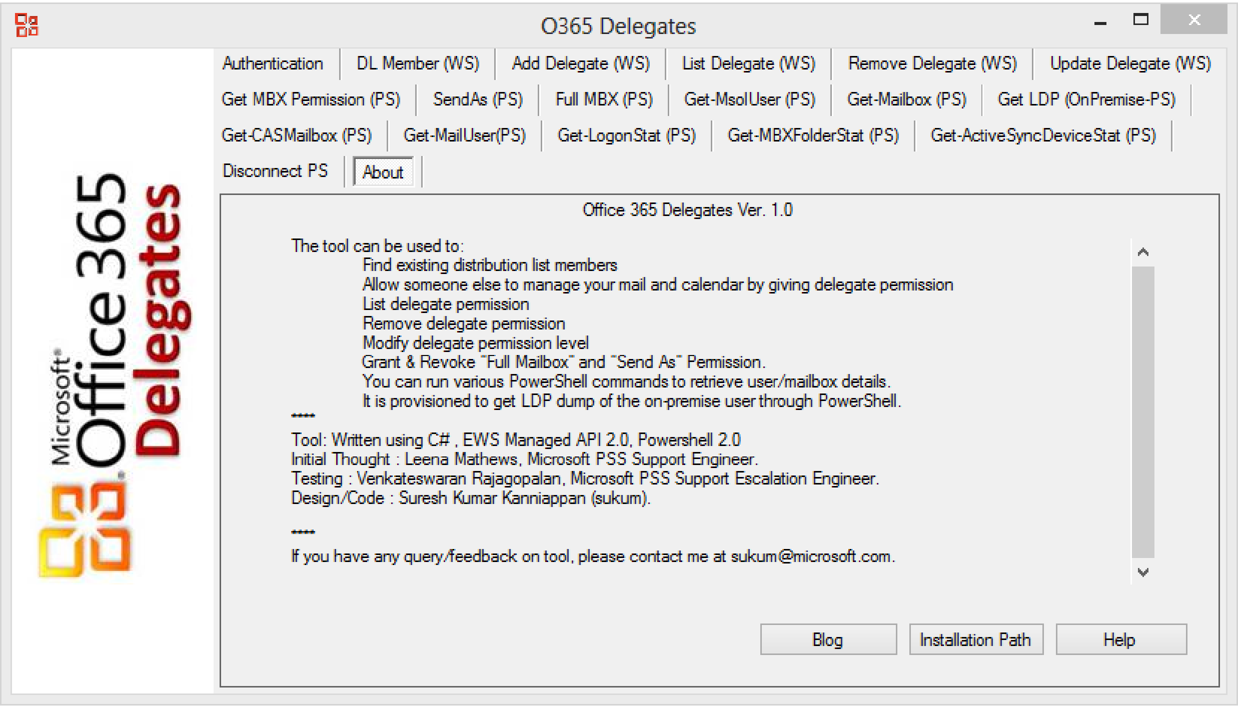
Go To Contents

Support/Feedback

If you have any questions or feedback, you can contact me at:

sukum@microsoft.com; or

k-sureshkumar@hotmail.com



Office 365 Delegates Ver. 1.0

The tool can be used to:
- Find existing distribution list members
- Allow someone else to manage your mail and calendar by giving delegate permission
- List delegate permission
- Remove delegate permission
- Modify delegate permission level
- Grant & Revoke "Full Mailbox" and "Send As" Permission.
- You can run various PowerShell commands to retrieve user/mailbox details.
- It is provisioned to get LDP dump of the on-premise user through PowerShell.

Tool: Written using C# , EWS Managed API 2.0, Powershell 2.0
Initial Thought : Leena Mathews, Microsoft PSS Support Engineer.
Testing : Venkateswaran Rajagopalan, Microsoft PSS Support Escalation Engineer.
Design/Code : Suresh Kumar Kanniappan (sukum).

If you have any query/feedback on tool, please contact me at sukum@microsoft.com.

Go To Contents