## Documentation for PowerShell Script: Distribution Group Management and Email Synchronization

This script automates the management of distribution groups in Microsoft Exchange Online. It compares current membership of various distribution groups to data sourced from CSV files and synchronizes membership by adding or removing users as necessary. Additionally, it generates log files for each action taken, such as added or removed users, and notifies administrators if specific CSV files are missing.

### **1. Overview**

The script does the following:

* Connects to Exchange Online using credentials.
* Imports CSV files containing the latest distribution group memberships.
* Compares the data to current members of specific distribution groups.
* Adds or removes members from the distribution groups based on the comparison.
* Logs added or removed members for each group.
* Sends notifications if a CSV file is missing or if the script runs into an error.

### **2. Variables Overview**

* **Credentials**: The script starts by defining the username and password to connect to Exchange Online using $Username, $PWord, and $Credential.
* **CSV File Locations**: All distribution group CSV files are located in a shared network folder. The path is stored in $CSVPath.
* **Group Names**: Variables like $DGP1Name, $DGP2Name, $DGP3Name, and others are used to identify various distribution groups (e.g., Announcements, Department Email, HR, etc.).
* **Log Files**: The script creates log files to track any changes made to the groups, such as additions or removals. Log file paths are dynamically created with timestamps.
* **Opt-Out and Volunteer Files**: The $DGP2OptOut and $Volunteer files hold user data for opt-out or volunteer-based membership management.

### **3. File and Folder Structure**

#### **CSV Files**

These CSV files contain the current list of members for each distribution group:

* **Location**: Wherever you would like to store them
* **File Format**: The files should match a specific naming pattern, for example:
  + \*DGP1Name\*yyyy-MM-dd.csv for the DGP1Name group

#### **Log Files**

These logs track changes to group memberships and are saved in:

* **Location**: Wherever you would like the to be stored
* **File Format**: Log files are named according to the action performed (Added, Removed), group name, and timestamp. Example:
  + Added DGP1Name 2024-01-25\_01\_20\_45.log
  + Removed DGP2Name 2024-01-25\_01\_20\_45.log

#### **OptOut File**

* **Location**: any file path to \DGP2OptOut.csv
* **Purpose**: Lists users who have opted out and should be removed from certain distribution groups.

#### **Volunteer File**

* **Location**: any file path to \Volunteers.csv
* **Purpose**: Contains email addresses of users who volunteer to be added to multiple distribution groups.

### **4. Script Breakdown**

#### **4.1 Exchange Online Connection**

$Username = "[Email address of account used for this action]"  
$PWord = ConvertTo-SecureString -String "[password]" -AsPlainText -Force  
$Credential = New-Object -TypeName System.Management.Automation.PSCredential -ArgumentList $Username, $PWord  
Connect-ExchangeOnline -Credential $Credential

* This connects the script to Exchange Online using the provided credentials.

#### **4.2 File Paths**

Variables such as $CSVPath, $DGP2OptOut, $Volunteer, and $Term define the paths and filenames for various CSV files and data related to distribution group memberships.

#### **4.3 Log File Creation**

Log files are dynamically generated using timestamps:

$DGP1Add = "$FileLocation\Added $DGP1Name $(get-date -f yyyy-MM-dd\_hh\_mm\_ss).log"  
$DGP1Rm = "$FileLocation\Removed $DGP1Name $(get-date -f yyyy-MM-dd\_hh\_mm\_ss).log"  
Out-File $DGP1Add  
Out-File $DGP1Rm

Each log file corresponds to a specific action (add or remove) for a given distribution group.

#### **4.4 Synchronization Process**

For each distribution group, the script performs the following:

1. **Read the current distribution group members** from Exchange Online using Get-DistributionGroupMember.
2. **Load the CSV file** containing the updated list of members.
3. **Add new members** from the CSV if they are not already in the group.
4. **Remove members** who are no longer present in the CSV file from the group.
5. **Update log files** with the list of added or removed members.

For example, for the Announcements group:

$DGP1NameCurrentList = Get-DistributionGroupMember -Identity $DGP1Name -ResultSize Unlimited  
$DGP1NameCSVList = import-csv $DGP1CSV | ForEach { $\_."x\_emails.int\_email" }  
  
$DGP1NameCurrentList | ForEach {  
 if ($DGP1NameCSVList -notcontains $\_) {  
 Remove-DistributionGroupMember -Identity $DGP1Name -Member $\_ -Confirm:$False  
 Add-Content $DGP1Rm $\_  
 }  
}  
  
$DGP1NameCSVList | ForEach {  
 if ($DGP1NameCurrentList -notcontains $\_) {  
 Add-DistributionGroupMember -Identity $DGP1Name -Member $\_ -Confirm:$False  
 Add-Content $DGP1Add $\_  
 }  
}

#### **4.5 Handling Missing Files**

If a required CSV file is missing, an email is sent to a system alarm address:

Send-MailMessage -SMTPServer [Mail server} -To [ destination email ] -From [Source email] -Subject "DGP1Name Distribution Group" -Body "csv for the DGP1Name distribution group is missing"

#### **4.6 Handling Volunteers**

Volunteers are added to multiple distribution groups. For each volunteer:

$Volunteers = import-csv $Volunteer  
$Volunteers | ForEach {  
 Add-DistributionGroupMember -Identity $DGP1Name -Member $\_ -Confirm:$False  
 Add-DistributionGroupMember -Identity $DGP1Name -Member $\_ -Confirm:$False  
 Add-DistributionGroupMember -Identity $DGP6Name -Member $\_ -Confirm:$False  
 Add-DistributionGroupMember -Identity $DGP7Name -Member $\_ -Confirm:$False  
 Add-Content $DGP1Add $\_  
 Add-Content $DGP2Add $\_  
 Add-Content $DGP6Add $\_  
 Add-Content $DGP7Add $\_  
}

#### **4.7 Opt-Out Management**

Users from the OptOut list are removed from the respective distribution group(s):

$OptOut = import-csv $DGP2OptOut  
$OptOut | ForEach {  
 Remove-DistributionGroupMember -Identity $DGP2Name -Member $\_ -Confirm:$False  
 Add-Content $DGP2Rm $\_  
}

### **5. Filename Format**

* **Distribution Group CSV Files**:

Naming format: \*<GroupName>\*yyyy-MM-dd.csv

Example for Announcements: \*Announce Report\*2024-01-25.csv

* **Log Files**:

Naming format: <Action> <GroupName> yyyy-MM-dd\_hh\_mm\_ss.log

Example: Added Announcements 2024-01-25\_01\_20\_45.log

### **6. Conclusion**

This script automates the synchronization of distribution group memberships in Exchange Online. It ensures that the members of the distribution groups are always up to date based on CSV files, logs the actions taken, and sends notifications in case of errors. The script requires proper organization of CSV files and appropriate permissions to access Exchange Online.