

Outlook Transition Script Setup Guide

This guide will detail the steps taken by Veterans Affairs Canada to setup an automated process for setting up new Outlook profiles after email migration happened.

Note: Veterans Affairs Canada is not responsible for anything going wrong. This is simply to share our solution.

Feel free to contact me at marc.arsenault@veterans.gc.ca.

How does it work?

The idea is to have a scheduled task run for users after their mailbox is migrated. The task runs a Powershell script that ultimately will wipe all existing Outlook profiles locally and setup the new profile using autodiscover.

The script won't continue until these 5 pre-checks are met:

1. User hasn't already migrated (checks reg key) – script disables itself if already migrated
2. Outlook is installed on the machine – script disables itself if Outlook not installed
3. Outlook has a profile for logged on user – script disables itself if no profile exist for user
4. Computer can access internal network (ping test)
5. User is part of a specific AD group (see Active Directory Setup section of guide)

Will users lose Outlook settings?

Because we are having to create a new Outlook profile for users, they will be losing mostly preference based settings such as default views. Signatures will be retained as they are not specific to profiles.

What will I require to set this up?

1. Access to create Group Policies
2. Access to a network share housing the Powershell script
3. An environment where users are using Outlook 2016 (script could probably be modified to work with other versions)
4. Powershell knowledge
5. Software to build .exe out of a .ps1 (We use Sapien's Powershell Studio 2019)
6. Access to create Active Directory groups

#1 Active Directory Setup

You will have to create two Active Directory groups.

1. Active Directory group to deploy the GPO
2. Active Directory group to populate with users who's mailbox has successfully migrated (needed in setting up script settings in the next section)

#2 Powershell Script Setup

You should have a copy of "OutlookTransitionScript.ps1". Open the script and modify the "Global Variables/Settings" section". After you are content with the settings, convert the .ps1 as a .exe that runs silently (no GUI option).

A few things to note:

1. \$Script:LoggingFolderPath must exist up to the last subdirectory. The last subdirectory will get created through running the script if it doesn't exist. Also make sure the user has access to write to that folder locally.
2. \$Script:MigrationADGroup must match with the AD group created earlier to house users who's mailboxes have successfully migrated. The script uses this to verify if it should be ran yet or not.
3. \$Script:MigrationScheduledTaskName must match exactly with the taskname created later through GPO

#3 GPO Setup

I'm going to assume if you're setting this up you know how to create and link a basic GPO.

Here are the settings we've used:

Script

Copy .exe for Outlook Transition Script to C:\Windows

Start menu shortcut (optional)

Create a start menu shortcut that opens the .exe script above. This provides a manual way of running the script for IT if need be.

Scheduled Task

Create a Scheduled Task that will run the Outlook Transition Script .exe on these triggers:

1. Every 5 minutes after specified date
2. At logon after specified date (optional)
3. On internal network connection event (optional)

We've used this XML to detect this (you'd replace prod.vac with your network profile name):

```
<QueryList> <Query Id="0" Path="System"> <Select Path="Microsoft-Windows-  
NetworkProfile/Operational">*[System[(EventID=10000)]] and  
*[EventData[(Data[@Name="Description"]="prod.vac")]] </Select> </Query> </QueryList>
```

Important things to note for scheduled task:

1. Must be run using user's context. (check checkbox "Run in logged-on user's security context (user policy option))
2. Must be using "Create" Action. This is because the script will disable the scheduled task on successful completion and we do not want it to be overwritten later and re-enabled.
3. Be sure to add an expiry date so it doesn't run indefinitely