© 2017 Soverance Studios
Scott McCutchen
Systems Administrator
scott.mccutchen@soverance.com
scott.mccutchen@bkv.com

Active Directory Reporting with Powershell

Script Documentation Version 1.0

Last Update: **05/25/2017**

The following documentation is designed to help system administrators run Active Directory reporting on their networks via Powershell. A complete AD Reporting script is included, and is designed to be run automatically on a schedule. Report information is consolidated into an Excel file, and then emailed for ease of access to end users and managers.

Pros:

- Built-in, cost-free, Microsoft-supported reporting solution for Windows environments
- No third parties, no installed software agents
- High level of granularity
- Scalable reporting can easily add or remove information
    - *If it exists on a computer, it can be added to the report.*
- Can be run manually or on a schedule

Cons:

- Will only report on Active Directory resources
- Arguably insecure method for storing SMTP password
    - *Use Option #2 when using Gmail SMTP for high security (see below)*
- Dependent on the Import-Excel module
    - *This dependency could be removed by exporting to CSV only*

Dependencies

This script requires the following software modules to be installed:

- **Powershell v2 or greater** (*installed by default in Windows*)
- **Microsoft Remote Server Administration Tools (RSAT)**
- **Powershell Import-Excel Module**

Powershell Version Info

Powershell is installed by default on all Windows machines, and will eventually replace the DOS-style command prompt (*it already does so on the latest Win10 machines*).

You can find the Powershell version of your current machine by entering the following command:

$PSVersionTable.PSVersion

Default Powershell versions for common Windows builds are listed below:

PS Version	Release Date	Operating System
Powershell 2.0	October 2009	Win 7, Server 2008 R2
Powershell 3.0	September 2012	Win 8, Server 2012
Powershell 4.0	October 2013	Win 8.1, Server 2012 R2
Powershell 5.0	April 2014	Win 10

Updating Powershell

Powershell can be updated to the latest version for most operating systems by installing the latest Windows Management Framework, and while it is not necessary to run this script, I highly recommend doing so (*especially on Windows 7 machines*). **The latest version of Powershell is version 5.1**.

Download and install Windows Management Framework 5.1 at the following URL:

https://www.microsoft.com/en-us/download/details.aspx?id=54616

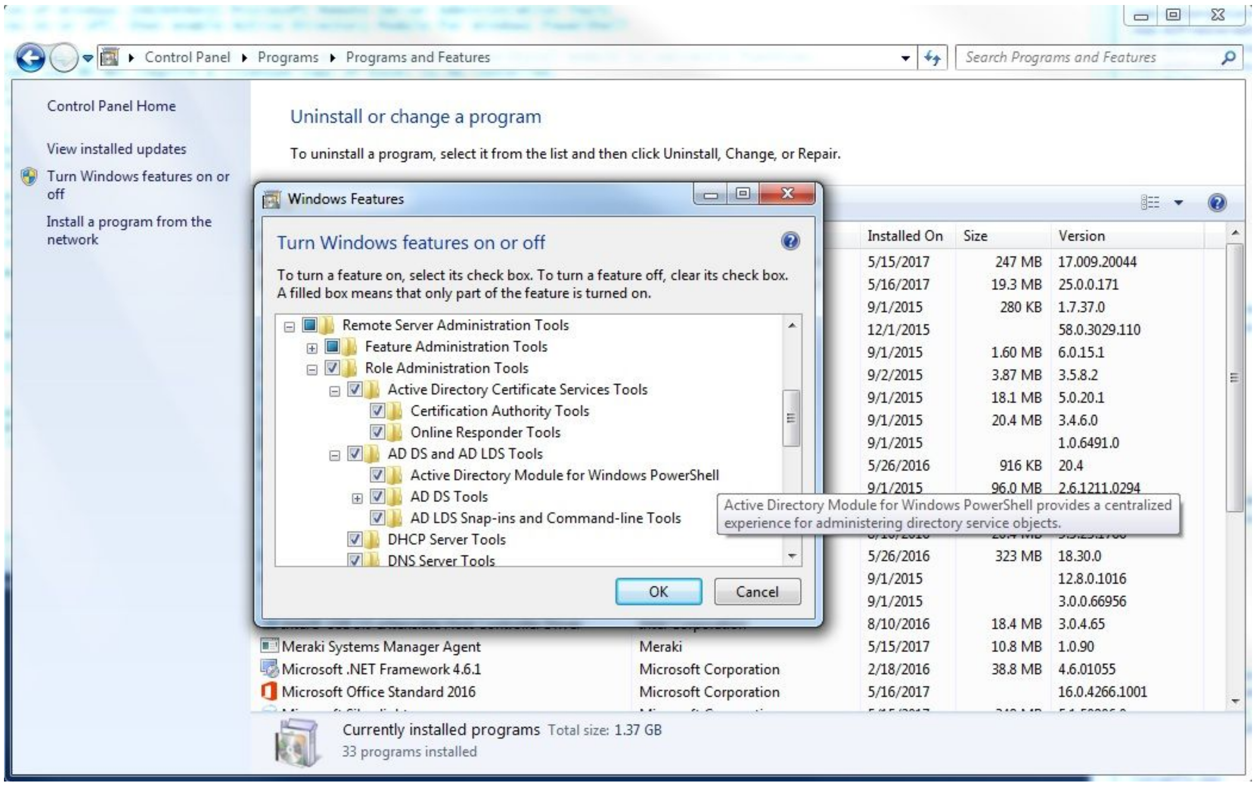
Microsoft RSAT Info

This script requires the **Active Directory Module for Windows Powershell**, available through the Microsoft Remote Server Administration Tools package. This module is installed by default on all Server 2008 and higher operating systems, but is not included in workstation builds (*such as 7 or 10*).

If you plan to run this script from your local workstation, you will need to download and install Microsoft RSAT package and enable the Active Directory Module for Windows Powershell.

RSAT for Windows 7: https://www.microsoft.com/en-us/download/details.aspx?id=7887
RSAT for Windows 10: https://www.microsoft.com/en-us/download/details.aspx?id=45520

Once RSAT is installed, navigate to through the Control Panel to “Add or Remove Programs” -> “Turn Windows Features on or off” -> and enable the Active Directory Module for Windows Powershell.



Control Panel ▸ Programs ▸ Programs and Features

Control Panel Home
View installed updates
Turn Windows features on or off
Install a program from the network

Uninstall or change a program

To uninstall a program, select it from the list and then click Uninstall, Change, or Repair.

Windows Features

Turn Windows features on or off

To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.

- ■ Remote Server Administration Tools
  - ■ Feature Administration Tools
  - ✓ Role Administration Tools
    - ✓ Active Directory Certificate Services Tools
      - ✓ Certification Authority Tools
      - ✓ Online Responder Tools
    - ✓ AD DS and AD LDS Tools
      - ✓ Active Directory Module for Windows PowerShell
      - ✓ AD DS Tools
      - ✓ AD LDS Snap-ins and Command-line Tools
    - ✓ DHCP Server Tools
    - ✓ DNS Server Tools

Active Directory Module for Windows PowerShell provides a centralized experience for administering directory service objects.

OK | Cancel

		Installed On	Size	Version
		5/15/2017	247 MB	17.009.20044
		5/16/2017	19.3 MB	25.0.0.171
		9/1/2015	280 KB	1.7.37.0
		12/1/2015		58.0.3029.110
		9/1/2015	1.60 MB	6.0.15.1
		9/2/2015	3.87 MB	3.5.8.2
		9/1/2015	18.1 MB	5.0.20.1
		9/1/2015	20.4 MB	3.4.6.0
		9/1/2015		1.0.6491.0
		5/26/2016	916 KB	20.4
		9/1/2015	96.0 MB	2.6.1211.0294
		5/26/2016	323 MB	18.30.0
		9/1/2015		12.8.0.1016
		9/1/2015		3.0.0.66956
		8/10/2016	18.4 MB	3.0.4.65
Meraki Systems Manager Agent	Meraki	5/15/2017	10.8 MB	1.0.90
Microsoft .NET Framework 4.6.1	Microsoft Corporation	2/18/2016	38.8 MB	4.6.01055
Microsoft Office Standard 2016	Microsoft Corporation	5/16/2017		16.0.4266.1001

Currently installed programs Total size: 1.37 GB
33 programs installed

Powershell Import-Excel Module

This script is designed to export directly to Excel .xlsx file format, and requires the Import-Excel module to do so. This dependency could be removed by exporting directly to CSV, but I’ve always preferred to work directly with Excel formats where possible to take advantage of Excel’s more powerful features.

With the Import-Excel module installed, you **DO NOT** require a licensed copy of Excel to be installed.

The Import-Excel module can be installed through the Powershell Gallery using the following commands (**PS v5 only!**):

To install for all users: Install-Module Import-Excel
To install only for current user: Install-Module Import-Excel -scope CurrentUser

PS v4 and below will need to follow installation instructions found on the module's Github page:
https://github.com/dfinke/ImportExcel

Further info on the Import-Excel module and various usage scenarios can be found in this blog post:
https://blogs.technet.microsoft.com/heyscriptingguy/2015/11/25/introducing-the-powershell-excel-module-2/

About Google (Gmail) SMTP

As Unified.Agency operates with Google For Work (G-Suite) email, we generally use Google's SMTP server to send automated email. Gmail is a highly secure environment, and requires additional configuration before Powershell can securely communicate with Gmail's SMTP server.

If you plan to send this report via Google SMTP servers, then you will need to enable one of the two following options on your Google account.

Option #1 - Allow Less Secure Apps

Visit this help document for more information and directions on how to enable this setting:

https://support.google.com/accounts/answer/6010255?hl=en

This option is not available for accounts with Two-Factor Authentication enabled (*use Option #2 instead*).

Allowing less secure apps to access your Google account is the easiest, but least secure method and could compromise the security of your account. I highly recommend using Option #2, below.

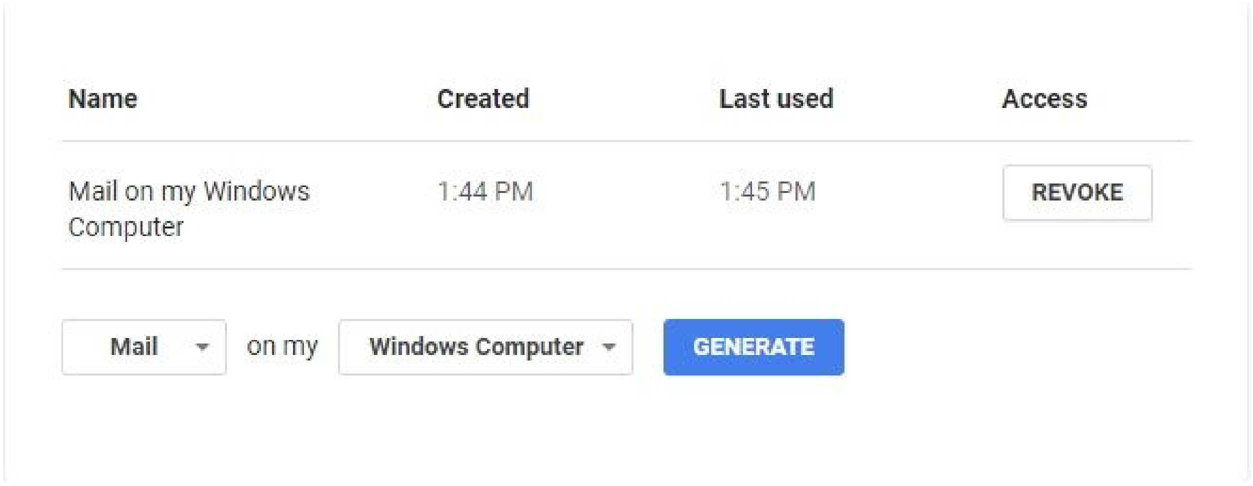
Option #2 - Use an "App Password"

Visit this help document for more information and directions on how to enable this setting:

https://support.google.com/accounts/answer/185833?hl=en

You must enable Two-Factor Authentication on your account before accessing this setting.

With Two-Factor Auth enabled, visit your App Passwords page (https://security.google.com/settings/security/apppasswords) and generate a new app password using the settings "Mail" and "Windows Computer", as shown below.



Name	Created	Last used	Access
Mail on my Windows Computer	1:44 PM	1:45 PM	REVOKE

Mail on my Windows Computer

You may then use this app password in place of your regular password when editing the script.

AD-Report.ps1 Usage

This script requires minor editing to be used in different environments. Before your first run of the script, you must hard-code the following information into the specified variables:

- **Domain Name**
    - *Variable Name:* $domain
    - *The name of the domain you wish to scan*
- **User**
    - *Variable Name:* $user
    - *The username of the account used to send SMTP mail*
- **Password**
    - *Variable Name:* $password
    - *The password of the account used to send SMTP mail*
- **SMTP Server**
    - *Variable Name:* $smtpserver
    - *The FQDN of the SMTP server*
- **SMTP Port**
    - *Variable Name:* $smtpport
    - *The port number required by the destination SMTP server*
- **Recipient**
    - *Variable Name:* $recipient
    - *The email address where you want the report sent to*

A quick note about security...

Ideally you are using a secure method to send this report via SMTP, using SSL (*port 465*) or TLS (*port 587*) encryption. Because the password is currently hard-coded into this script, using an insecure SMTP port (*such as port 25*) will expose your mail account to higher security risks. You can mitigate or eliminate this risk by using the "app password" feature of Gmail (*see Option #2, above*), and by following best practices in regards to folder permissions wherever this script is located.

Run the script manually

To run this script manually, open Powershell **as an administrator** and set the Execution Policy to allow for unsigned scripts to run on your system by issuing the following command:

Set-ExecutionPolicy RemoteSigned

Hit the "Y" key when prompted to set the execution policy. Then navigate to the directory where this script is located using the **Set-Location** command with a **Path** parameter, as shown below.

Set-Location -Path C:\Users\scottm\Documents\

Finally, run the script manually with the following command:

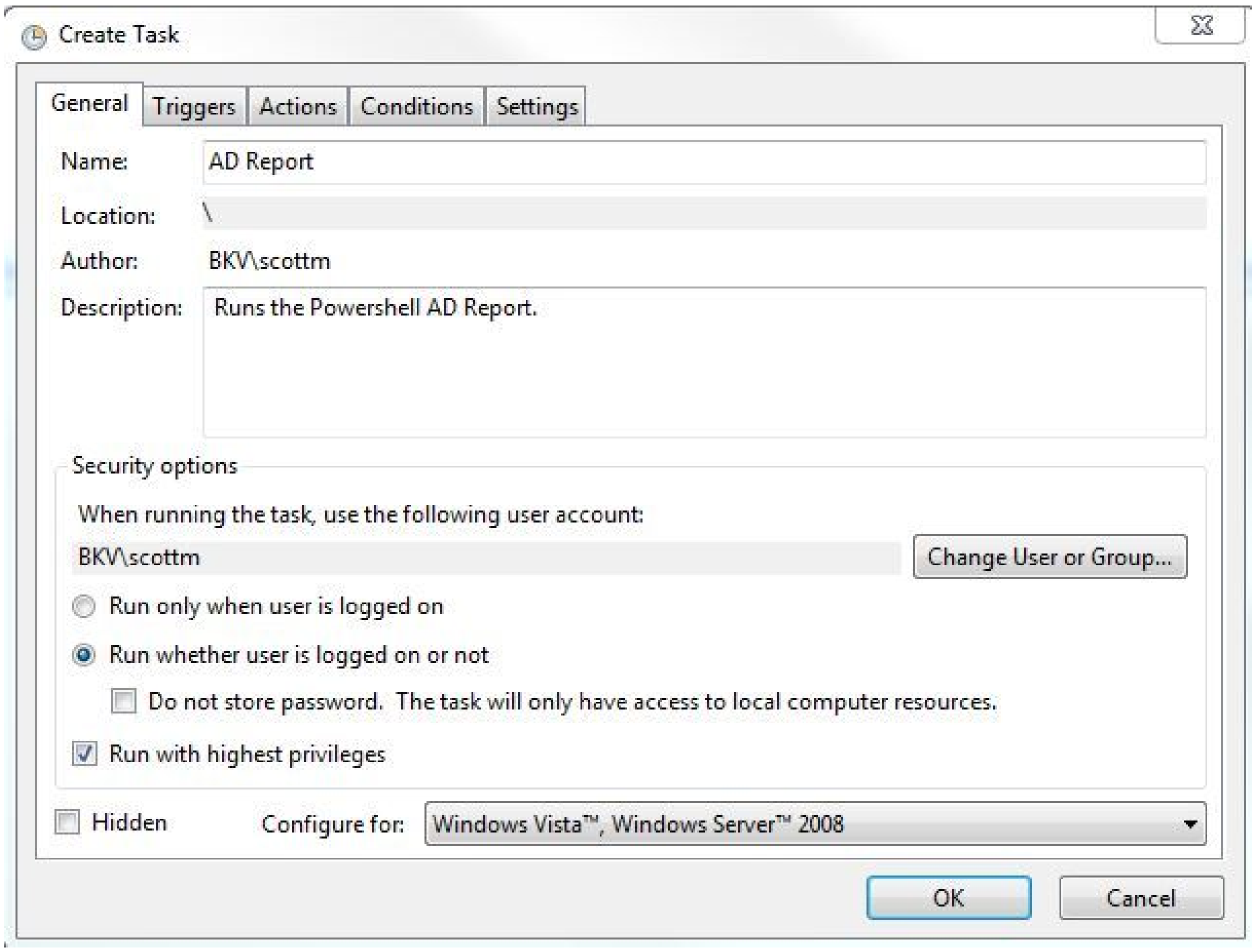
./AD-Report.ps1

Run the script automatically as a scheduled task

This script is designed to be run automatically as a scheduled task, using the Windows Task Scheduler.

To run this script on a recurring schedule, open Task Scheduler and configure the task.

First, create a new task using a descriptive name, and provide the task with administrator credentials. Run the task with highest privileges, and run it whether the user is logged on or not.



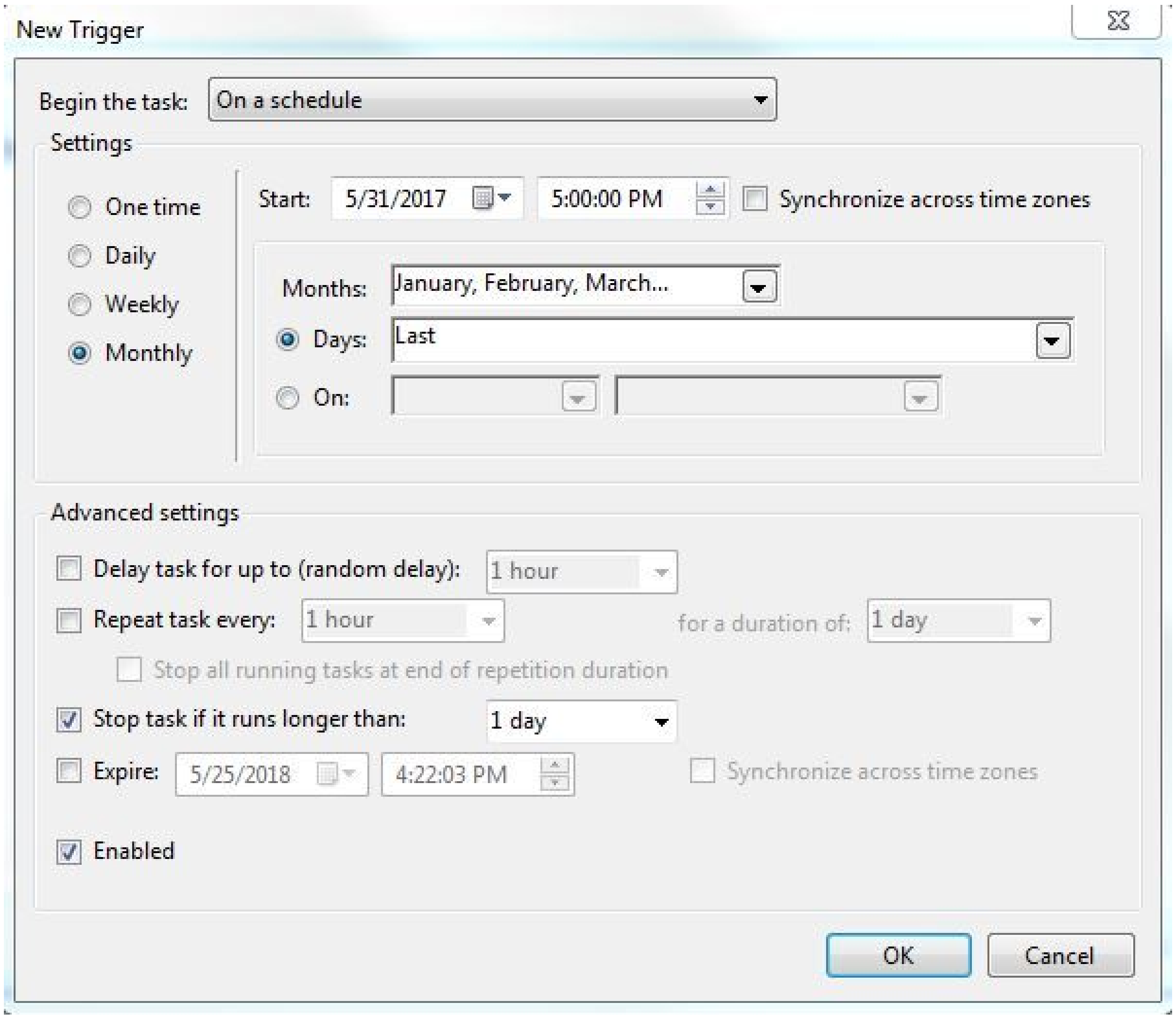
Name: AD Report
Location: \
Author: BKV\scottm
Description: Runs the Powershell AD Report.

Security options
When running the task, use the following user account:
BKV\scottm

○ Run only when user is logged on
● Run whether user is logged on or not
☐ Do not store password. The task will only have access to local computer resources.
☑ Run with highest privileges

☐ Hidden   Configure for: Windows Vista™, Windows Server™ 2008

Second, set a new trigger for this task. Begin the task on a schedule and set your parameters as necessary. In the screenshot below, I have specified the task to run once per month, on the last day of the month. You can set it to run more often if you like. Be sure to stop the task if takes too long, and be sure to check the “Enabled” box to allow this trigger to fire.


New Trigger

Begin the task: On a schedule

Settings

One time [ ] Daily [ ] Weekly [ ] [✓ Monthly]

Start: 5/31/2017 5:00:00 PM [ ] Synchronize across time zones

Months: January, February, March...

[✓] Days: Last

[ ] On:

Advanced settings

[ ] Delay task for up to (random delay): 1 hour

[ ] Repeat task every: 1 hour for a duration of: 1 day

[ ] Stop all running tasks at end of repetition duration

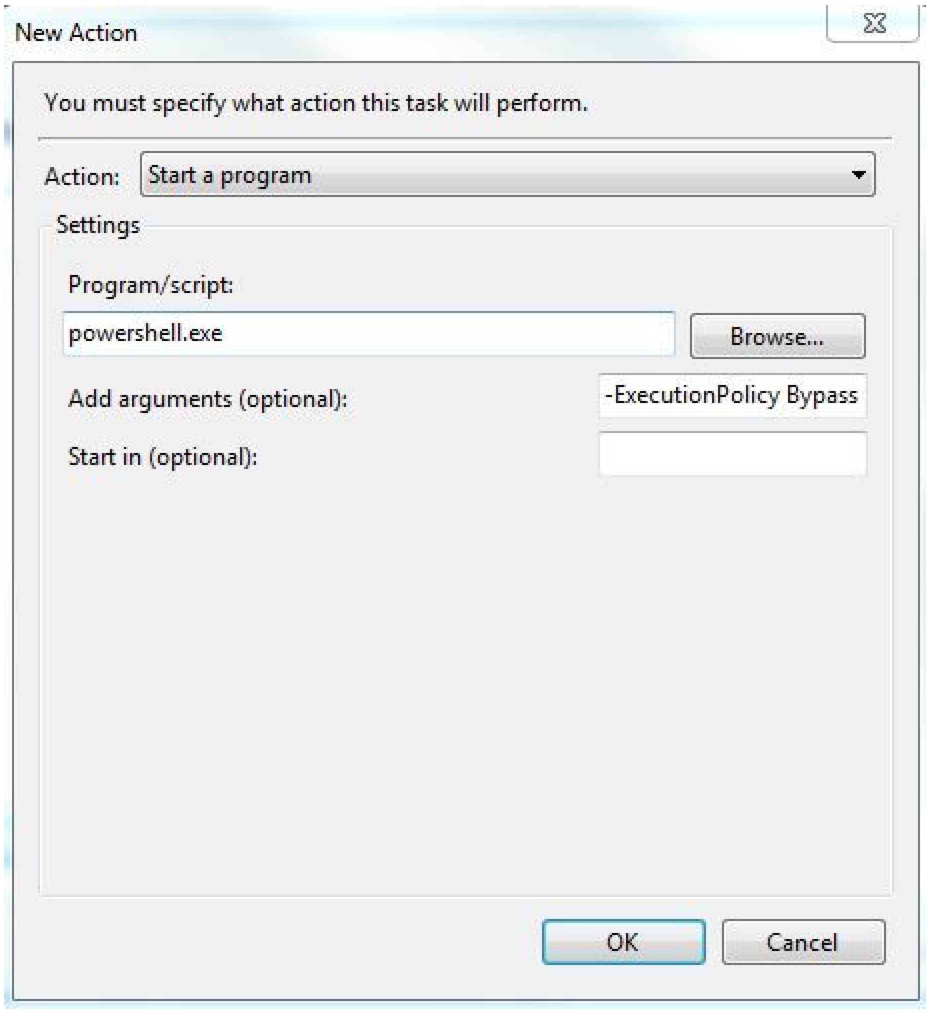
[✓] Stop task if it runs longer than: 1 day

[ ] Expire: 5/25/2018 4:22:03 PM [ ] Synchronize across time zones

[✓] Enabled

Third, specify a new action to "Start a program". Here, you'll want to start powershell.exe.

Additionally, you'll need to add two arguments - the first being the Execution Policy Bypass, and the second being the full path (*including filename*) of the AD Report script.



New Action

You must specify what action this task will perform.

Action: Start a program

Settings

Program/script: powershell.exe

Add arguments (optional): -ExecutionPolicy Bypass

Start in (optional):

Your full arguments line will look something like this:

-ExecutionPolicy Bypass C:\Users\scottm\Documents\AD-Report.ps1

Finally, you can leave the "Conditions" and Settings" tabs with their default values.

Save the task, and you're done. The task will now run as scheduled, and your AD report will be emailed when complete.