



1001 – Data Extracts & Automation

Wednesday, 03/06/2024
IT Day - Period 2
09:15 AM – 10:15 AM



Robert Cranston |
Integration Engineer



Sarah Walpole |
Integration Engineer

Agenda

01 Intro to PowerShell

02 Exporting Student Data

03 Variables

04 Templates

05 SQL Account

06 Logging

07 Batch Files

08 Automation



How do you export/import data?

- SSMS/SSIS/SSRS
- Python
- Other?
- PowerShell (ISE/Core)



What is PowerShell?

“PowerShell is a cross-platform task automation solution made up of a command-line shell, a scripting language, and a configuration management framework.” <https://learn.microsoft.com/en-us/powershell/scripting/overview?view=powershell-7.3>



Windows PowerShell ISE comes pre-installed on Windows 7, Server 2008, and every release after that. The last version is 5.1

PowerShell Core is open source and can be installed on MAC or Linux OS. The latest version of PowerShell Core is 7.3.9

Visual Studio Code is the editor of choice for PowerShell 7 functionality



AERIESCON



One of our schools just started using ABCXYZ Learning Program and they need student data immediately.

Exporting Student Data

- **Gather requirements from Vendor**

- File layout templates, SFTP information, Automation Schedule, Sample File, etc.

- **Define Variables**

- Set Variables for SQL Credentials, SFTP credentials, and Input/Output paths.

- **Create and Test SQL Script(s)**

- Prior to testing your script in PowerShell, have a functioning/well tested SQL script.

- **Batch Files**

- Can be used for both single and multiple file processing but are not necessary.

School_id	Student_id	Last_name	First_name
13K123	334494781	Potter	Harry
13K123	299786638	Weasley	Ron
13K123	819021071	Granger	Hermione
13K123	659578830	Diggory	Cedric
13K123	745029534	Weasley	George
13K123	660345149	Weasley	Fred



Variables

- **SQL Variable Examples**

```
$SQLUser = "username"  
$SQLPass = "password"  
$SQLServer = "servername(FQDN)"  
$BasePath = "C:\Powershell\  
$FileName = "students.csv"  
$ExcludedSchools = "999"
```

- **Using SQL Variables in PowerShell:**

```
Invoke-Sqlcmd -serverinstance $SQLServer -database $SQLDB -username $SQLUser -password $SQLPass -query "
```

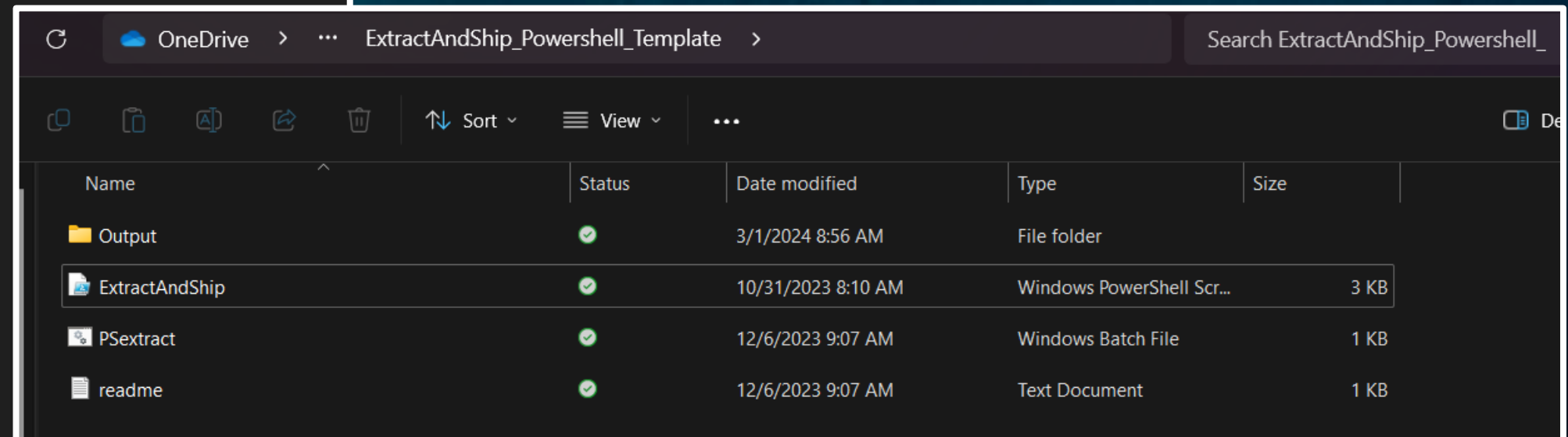
```
--SELECT * FROM STU WHERE SC NOT IN $ExcludedSchools
```

```
" | Export-Csv -NoTypeInfoInformation -Path $OutPath\$FileName
```



Start with a Template

```
1 #Extract file(s) and send via SFTP
2 #Automatically use the most recent db given the suffix
3
4 Clear-Host
5
6 $SQLUser = "username"
7 $SQLPass = "password"
8 $SQLServer = "servername(FQDN)"
9 $OutPath = "C:\Scripts\DISTRICT\EXTRACTNAME\Output\"
10 $FileName = "students.csv" #Define for one file or individually for multiple files below
11 $DBSuffix = "AeriesDemo"
12
13 $currentYearDB =
14     Invoke-Sqlcmd -serverinstance $SQLServer -database master -username $SQLUser -password $SQLPass -TrustServerCertificate -query "
15         DECLARE @SUFFIX VARCHAR(60) = '$DBSuffix';
16         SELECT top(1) [NAME] FROM SYS.DATABASES WHERE [NAME] LIKE '%' + @SUFFIX AND [NAME] NOT LIKE '%[_]%'
17         ORDER BY [NAME] DESC
18     "
19 $SQLDB = $currentYearDB.NAME
20
21
22 #SFTP server info
23 $doSFTP = 0 #Enable/Disable SFTP 0=no, 1=yes
24 $strSN = "SFTPserver" #SFTP Server Name or Static IP
25 $strUN = "SFTPusername" #SFTP UserName
26 $strPW = "SFTPpassword" #SFTP Password
27 $strRD = "RemoteDirectory" #SFTP Remove Directory
28 $sshKey = "ssh-rsa 1024 ..." #SSH Key
29
30 # Load WinSCP .NET assembly
31 Add-Type -Path "C:\WinSCP\WinSCPnet.dll"
32 $MyDir = [System.IO.Path]::GetDirectoryName($myInvocation.MyCommand.Definition)
33 Set-Location $mydir
34
35
36 if ($null -ne $SQLDB) {
37
38     Invoke-Sqlcmd -serverinstance $SQLServer -database $SQLDB -username $SQLUser -password $SQLPass -TrustServerCertificate -query "
39
40     --insert SQL statement here
41
42
43     " | Export-Csv -NoTypeInformation -Path $OutPath\$filename
44 }
```



Name	Status	Date modified	Type	Size
Output	✓	3/1/2024 8:56 AM	File folder	
ExtractAndShip	✓	10/31/2023 8:10 AM	Windows PowerShell Scr...	3 KB
PSextract	✓	12/6/2023 9:07 AM	Windows Batch File	1 KB
readme	✓	12/6/2023 9:07 AM	Text Document	1 KB



AERIESCON

SQL Security and Script

- Create Individual SQL login and Security Role for each product
- Write and test your SQL script



Populate your PowerShell and Run

- Fill in template variables
- Run script without SFTP

```
20
21
22  #SFTP server info
23  $doSFTP = 0           #Enable/Disable SFTP 0=no, 1=yes
24  $strSN  = "SFTPserver" #SFTP Server Name or Static IP
25  $strUN  = "SFTPusername" #SFTP UserName
26  $strPW  = "SFTPpassword" #SFTP Password
27  $strRD  = "RemoteDirectory" #SFTP Remove Directory
28  $sshKey = "ssh-rsa 1024 ..." #SSH Key
29
```



Logging

Add logging to your PowerShell script to display progress of your extract. Logs are useful when issues arise with your extract, and you need to find the breakpoint.

Logging commands for beginning of script:

Start-Transcript - "C:\Path\To\log.txt"

Logging command for end of script:

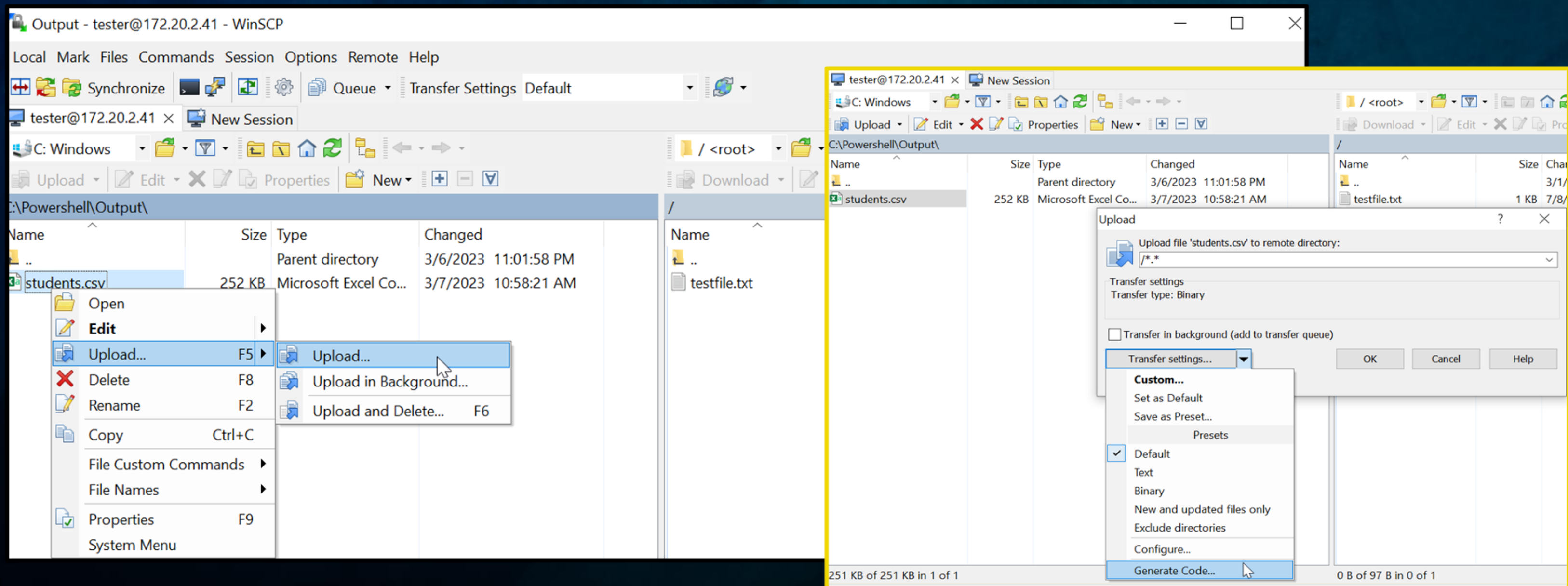
Stop-Transcript

You can use **Write-Host** to set verbiage that will be added to the log.



Use WinSCP to generate PowerShell code to send data via SFTP

In WinSCP, connect to the destination SFTP folder as well as the Origin folder. Select the file/s to be uploaded, right-click on the file/s, select Upload... twice. Then select Generate Code Transfer setting.

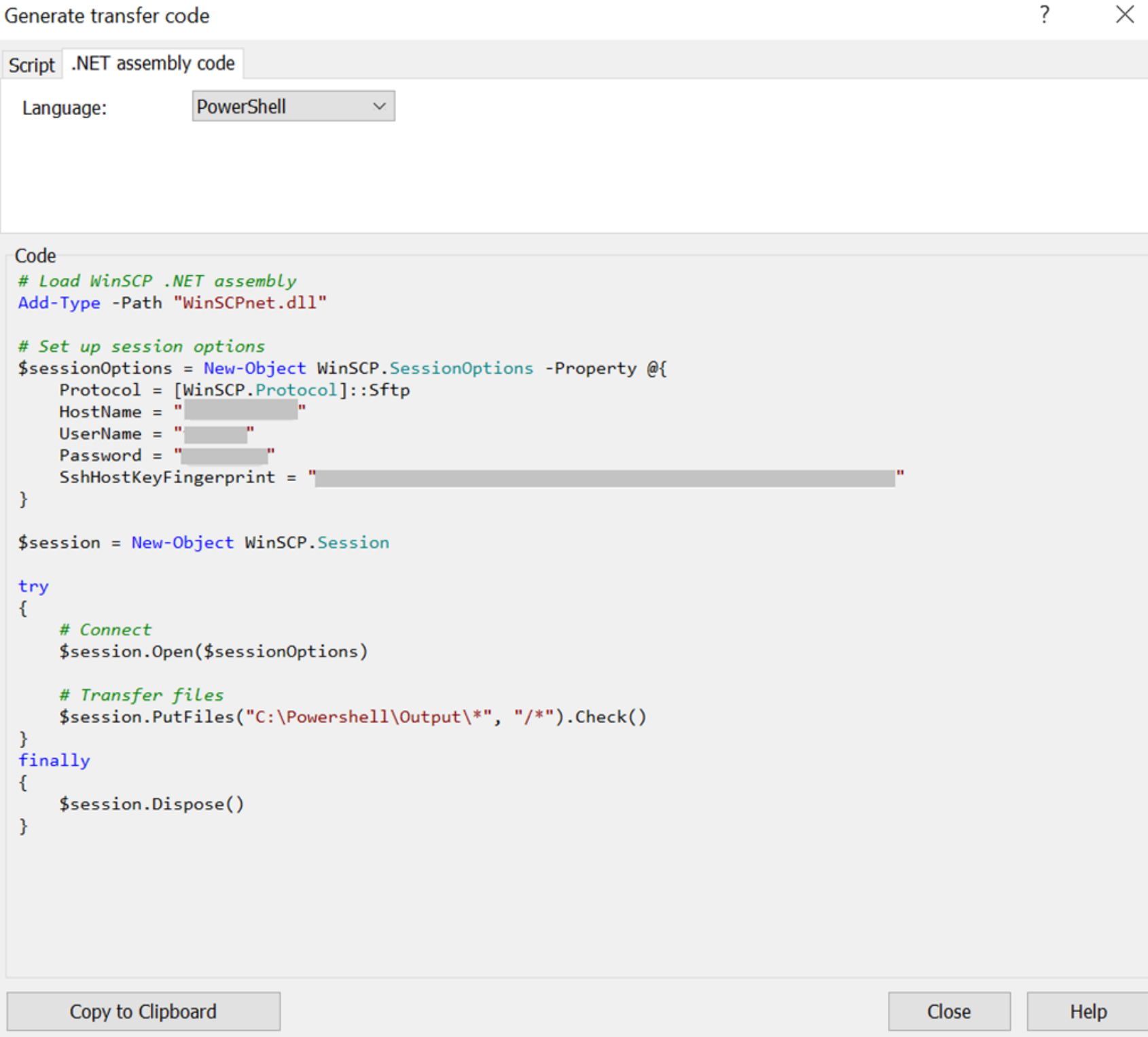


WinSCP Functionality

Use the .NET assembly code tab and select PowerShell from the Language dropdown.

Select Copy to Clipboard to copy code to your file.

You will need to update the Path to where WinSCPnet.dll is located.



```
Generate transfer code
Script .NET assembly code
Language: PowerShell

Code
# Load WinSCP .NET assembly
Add-Type -Path "WinSCPnet.dll"

# Set up session options
$sessionOptions = New-Object WinSCP.SessionOptions -Property @{
    Protocol = [WinSCP.Protocol]::Sftp
    HostName = "
    UserName = "
    Password = "
    SshHostKeyFingerprint = "
}

$session = New-Object WinSCP.Session

try
{
    # Connect
    $session.Open($sessionOptions)

    # Transfer files
    $session.PutFiles("C:\Powershell\Output\*", "/").Check()
}
finally
{
    $session.Dispose()
}
```



Batch Files

Create batch files (.bat) to use in processing your files.

Sample batch file code:

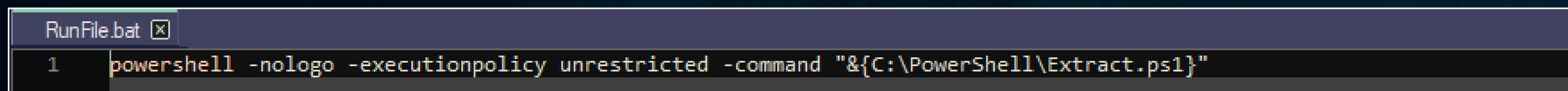
powershell is the program if using PowerShell 5, pwsh is the program if using PowerShell 7

- nologo - hides the copyright banner at startup.

- executionpolicy - sets execution policy

- command – In our scenario, the value passed to the Command is a string and is using the call operator “&” to execute the inline script block defined inside our string which is our PowerShell script.

Insert Batch file article link here



```
1 powershell -nologo -executionpolicy unrestricted -command "&{C:\PowerShell\Extract.ps1}"
```



Methods of Automation

- What do you use to Automate?

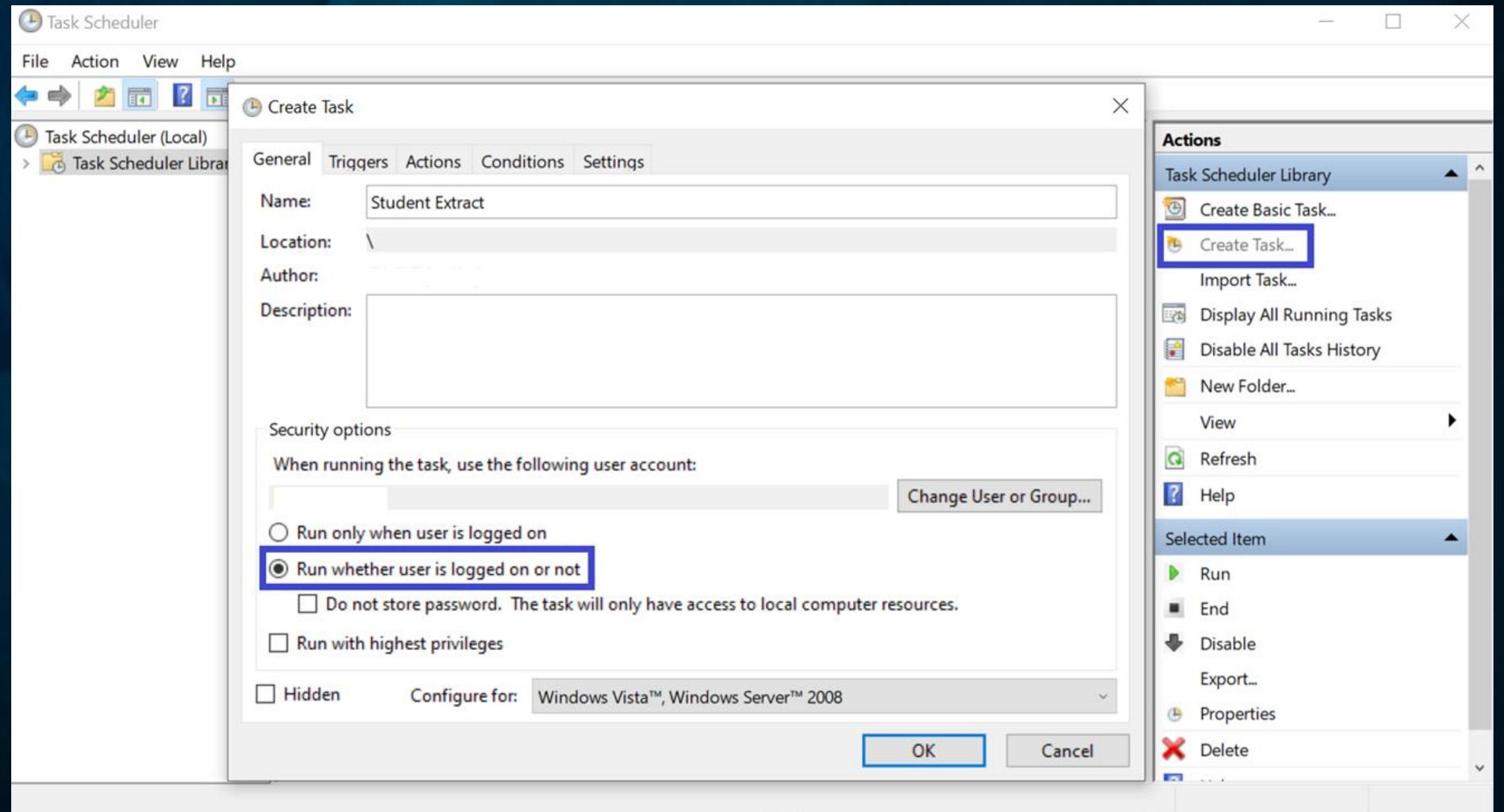


AERIESCON

Use Task Scheduler to automate your extract

Create Task

- Open Task Scheduler
- Create Task
- Add Name
- Set to Run whether user is logged on or not



AERIESCON

Task Scheduler - Triggers

Create Task

General Triggers Actions Conditions Settings

When you create

Trigger

New Trigger

Begin the task: On a schedule

Settings

☐ One time

☐ Daily

☒ Weekly

☐ Monthly

Start: 3/ 7/2023 1:07:01 PM ☐ Synchronize across time zones

Recur every: 1 weeks on:

☐ Sunday ☒ Monday ☒ Tuesday ☒ Wednesday

☒ Thursday ☒ Friday ☐ Saturday

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 hour

☐ Expire: 3/ 7/2024 1:07:06 PM ☐ Synchronize across time zones

☒ Enabled

When running the task

INTERACTIVE

☒ Run only when user is logged on

☐ Run whether or not user is logged on

☐ Do not stop the task if the logon session ends

New...

OK Cancel

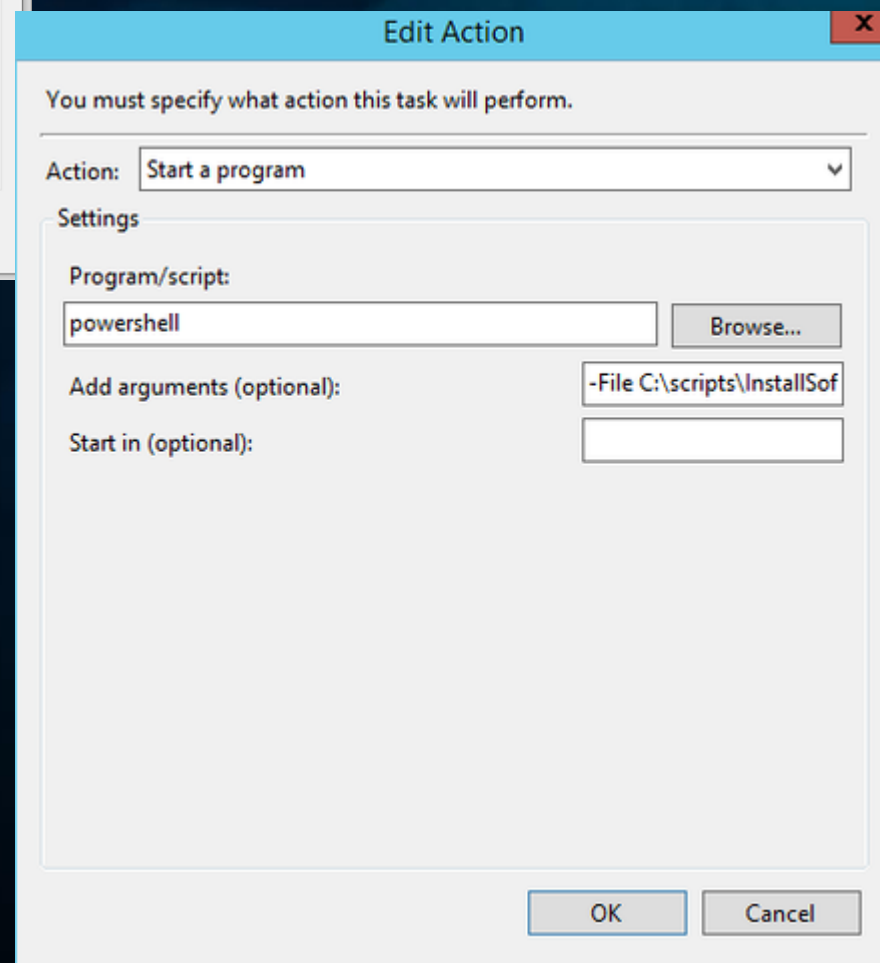
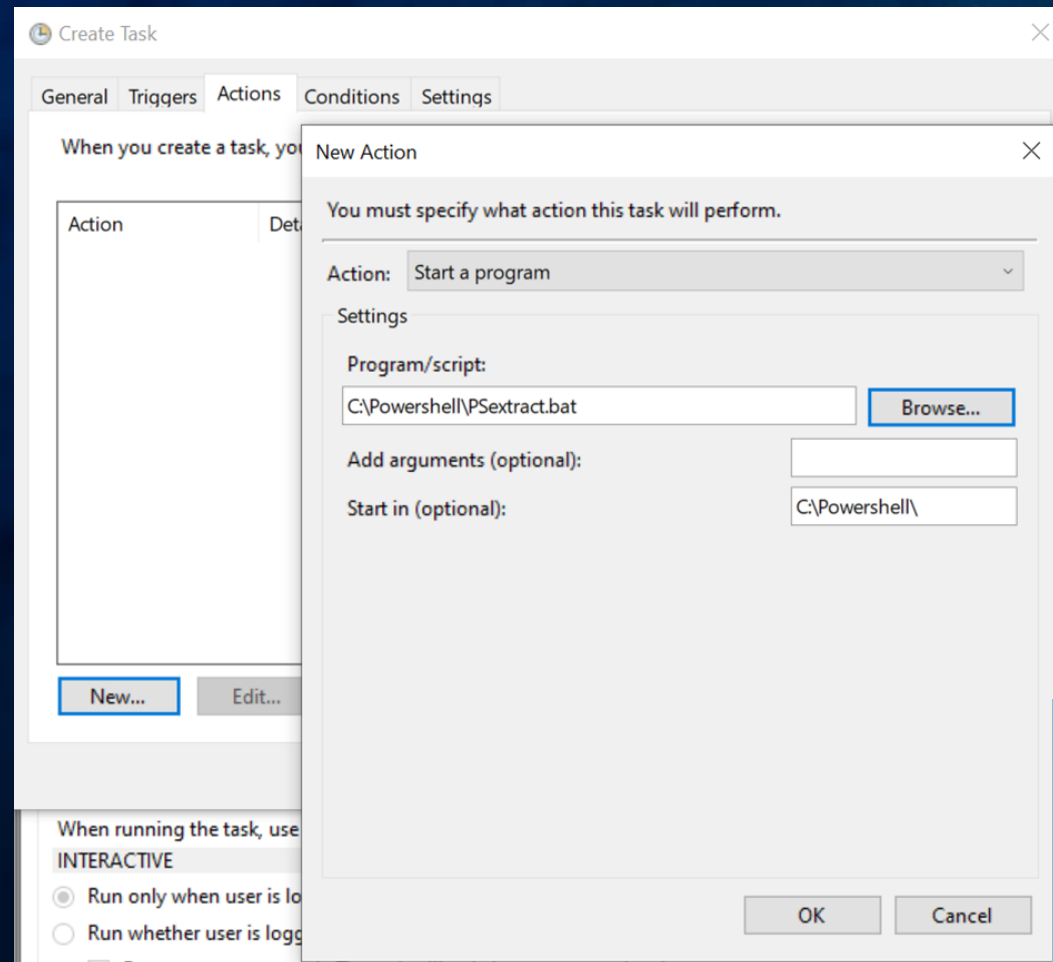
Triggers

- New Trigger
- Set Schedule
- Set Stop task if it runs longer than to 3 hours or less.



AERIESCON

Task Scheduler - Actions



Actions

- New Action
- Action should be Start a Program
- Set Path for Program/script
- CHANGE THESE



AERIESCON

Task Scheduler – Conditions and Settings

Conditions and Settings Tabs

- These options will be determined by your District policies.

General Triggers Actions **Conditions** Settings History

Specify the conditions that, along with the trigger, determine whether the task should run. The task will not run if any condition specified here is not true.

Idle

☐ Start the task only if the computer is idle for: 10 minutes

Wait for idle for: 1 hour

☒ Stop if the computer ceases to be idle

☐ Restart if the idle state resumes

Power

☐ Start the task only if the computer is on AC power

☒ Stop if the computer switches to battery power

☐ Wake the computer to run this task

Network

☒ Start only if the following network connection is available:

Any connection

OK Cancel

General Triggers Actions **Conditions** Settings History

Specify additional settings that affect the behavior of the task.

☒ Allow task to be run on demand

☒ Run task as soon as possible after a scheduled start is missed

☐ If the task fails, restart every: 1 minute

Attempt to restart up to: 3 times

☒ Stop the task if it runs longer than: 1 day

☒ If the running task does not end when requested, force it to stop

☐ If the task is not scheduled to run again, delete it after: 30 days

If the task is already running, then the following rule applies:

Do not start a new instance

OK Cancel



THANK YOU!

Feedback:



Please take a moment to complete our session survey.
<http://aeries.AeriesCon-Session-Feedback-Survey-Spring-2024.alchemer.com/s3/>



AERIESCON