Search Guide

This PowerShell utility was designed to search files quickly and accurately for matches. The default (Search.ps1) will report files containing "-COMPUTERNAME" (Your PC name) in your OneDrive. OneDrive creates these files when there is some sort of synchronization issue. In theory, this OneDrive feature preserves your work so you can determine which to keep.

I have not found this helpful. I have spent a lot of time cleaning up after these sync errors. Now, to facilitate this process, I have this script to report these duplicates. However, the utility can be modified to do searching (optionally deleting) using any kind of search term and/or multiple locations. Like searching for duplicates in any number of EverQuest install folders (e.g., "C:\Users\Public\Daybreak Game Company\Installed Games", where there are multiple subdirectories for Live, Test, Beta…). These duplicates can happen when you are synchronizing configurations to OneDrive (e.g., SyncMulti.bat utility).

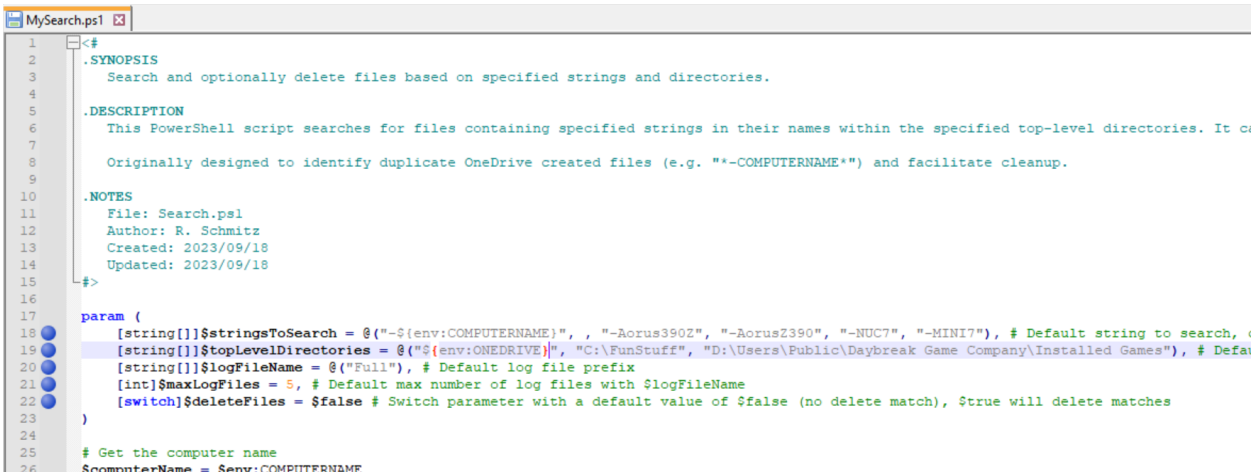
Usage

Generally, this should be run as needed. Although you could set up a task to run it automatically. Included in the download are several scripts to demonstrate different uses. The default, Search.ps1, searches (report only) files containing "-COMPUTERNAME" (Your PC Name) which are likely duplicates created by OneDrive. The parameters are shown below:

Search.ps1

```
<#
.SYNOPSIS
    Search and optionally delete files based on specified strings and directories.

.DESCRIPTION
    This PowerShell script searches for files containing specified strings in their names within the specified top-level directories.

    Originally designed to identify duplicate OneDrive created files (e.g. "*-COMPUTERNAME*") and facilitate cleanup.

.NOTES
    File: Search.ps1
    Author: R. Schmitz
    Created: 2023/09/18
    Updated: 2023/09/18
#>

param (
    [string[]]$stringsToSearch = @("-${env:COMPUTERNAME}"), # Default string to search, comma-delimited
    [string[]]$topLevelDirectories = @("${env:ONEDRIVE}"), # Default top-level directory(ies) to recursively search, comma-delimited
    [string[]]$logFileName = @("Search"), # Default log file prefix
    [int]$maxLogFiles = 5, # Default max number of log files with $logFileName
    [switch]$deleteFiles = $false # Switch parameter with a default value of $false (no delete match), $true will delete matches
)

# Get the computer name
```

Other samples include broader search terms and locations, such as one found when synchronizing EverQuest configuration files to OneDrive. Example of a customized multiple term and location without delete below:

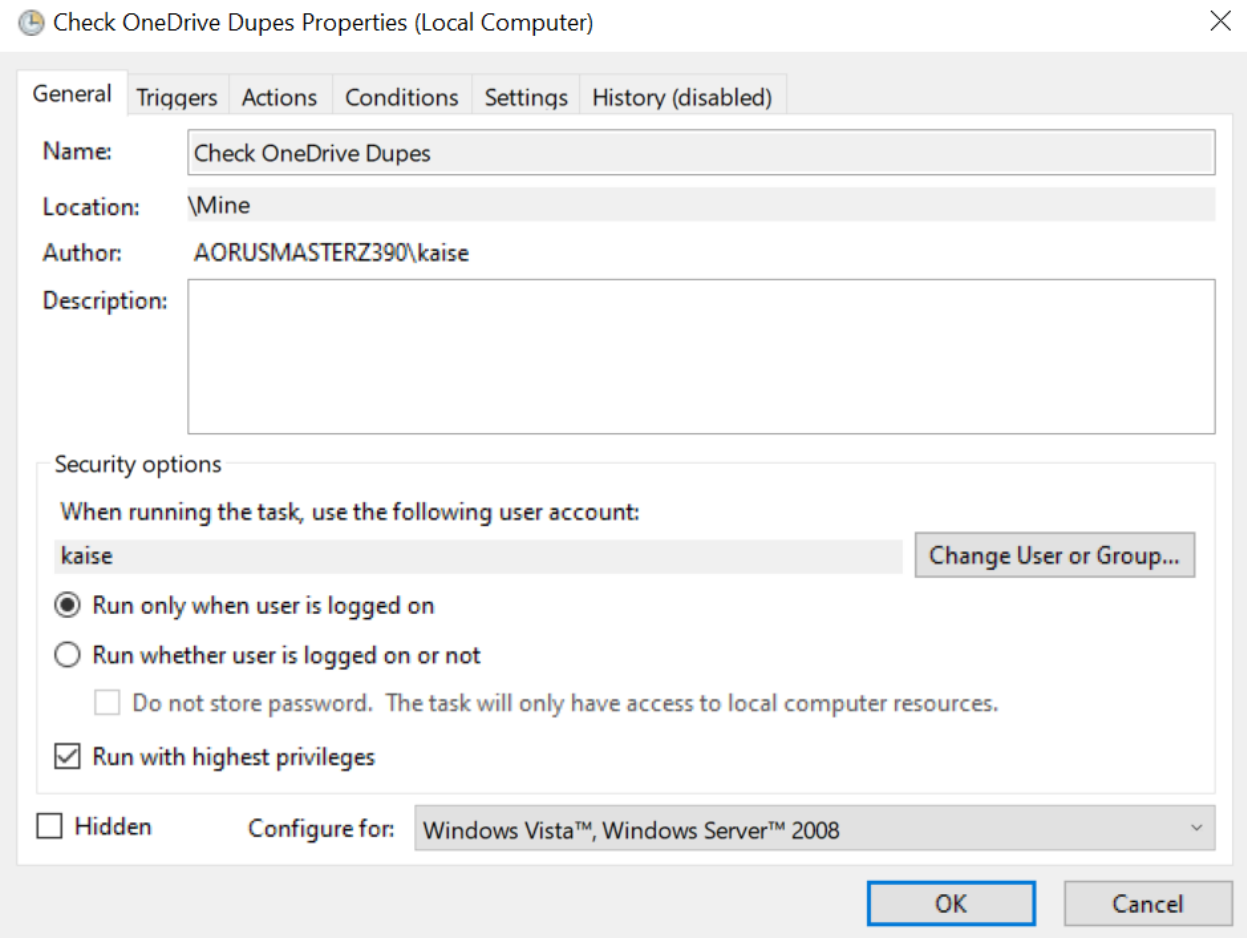


Task Scheduler (Option)

Task Scheduler option for executing PowerShell scripts takes a bit of extra work. The steps are below in case it is something you wish to setup.

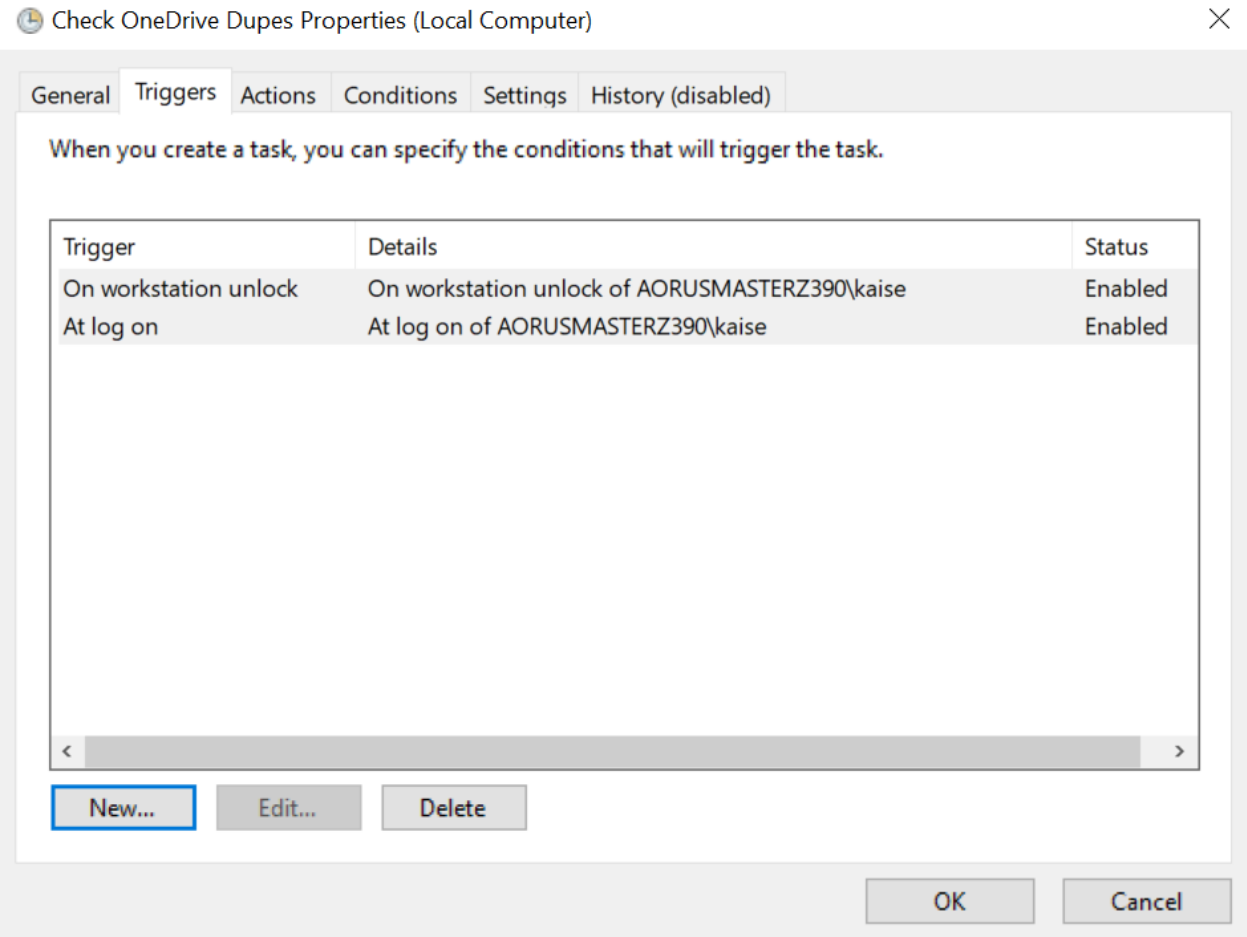
NOTE: I have included samples for scripts which check if eqgame.exe is running and if OneDrive is currently syncing or just check if OneDrive is syncing. These can be used in Task Scheduler to kick off these other utilities (Search.ps1, MySearch.ps1, SyncMulti.bat, etc). The checks for eqgame.exe and OneDrive should help alleviate duplicates/sync issues. It is smart enough to sleep and check again (currently defined to be 5 minutes). So, Task Scheduler can still be set for Log On / Unlock / Lock etc (or more often) without any worries. It uses a lock and timestamp file to ensure only one maximum is running (as defined by the lock and timestamp file names).

General



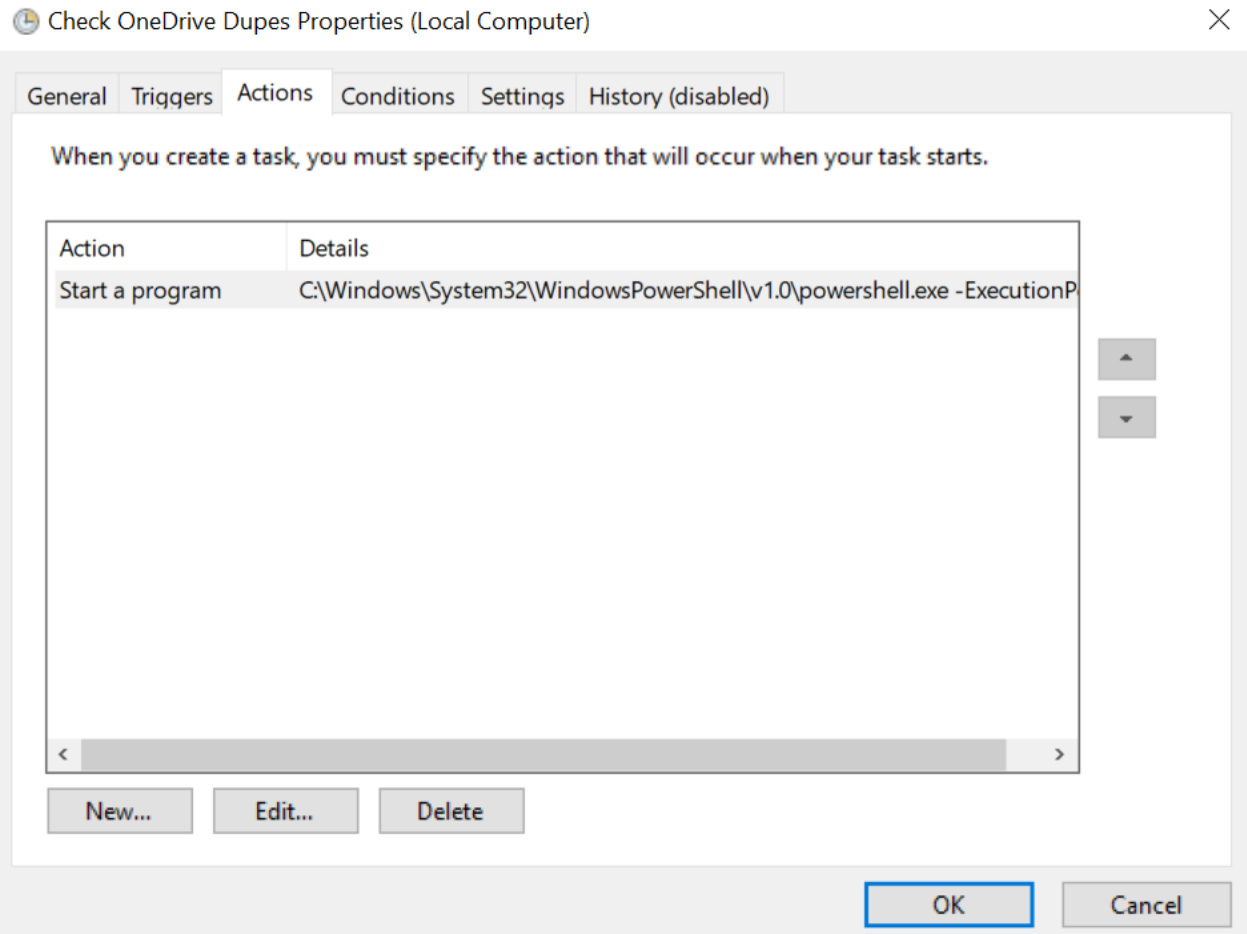
Give it a name and check the box for highest privileges. I set this one to run only when my user is logged in.

Triggers

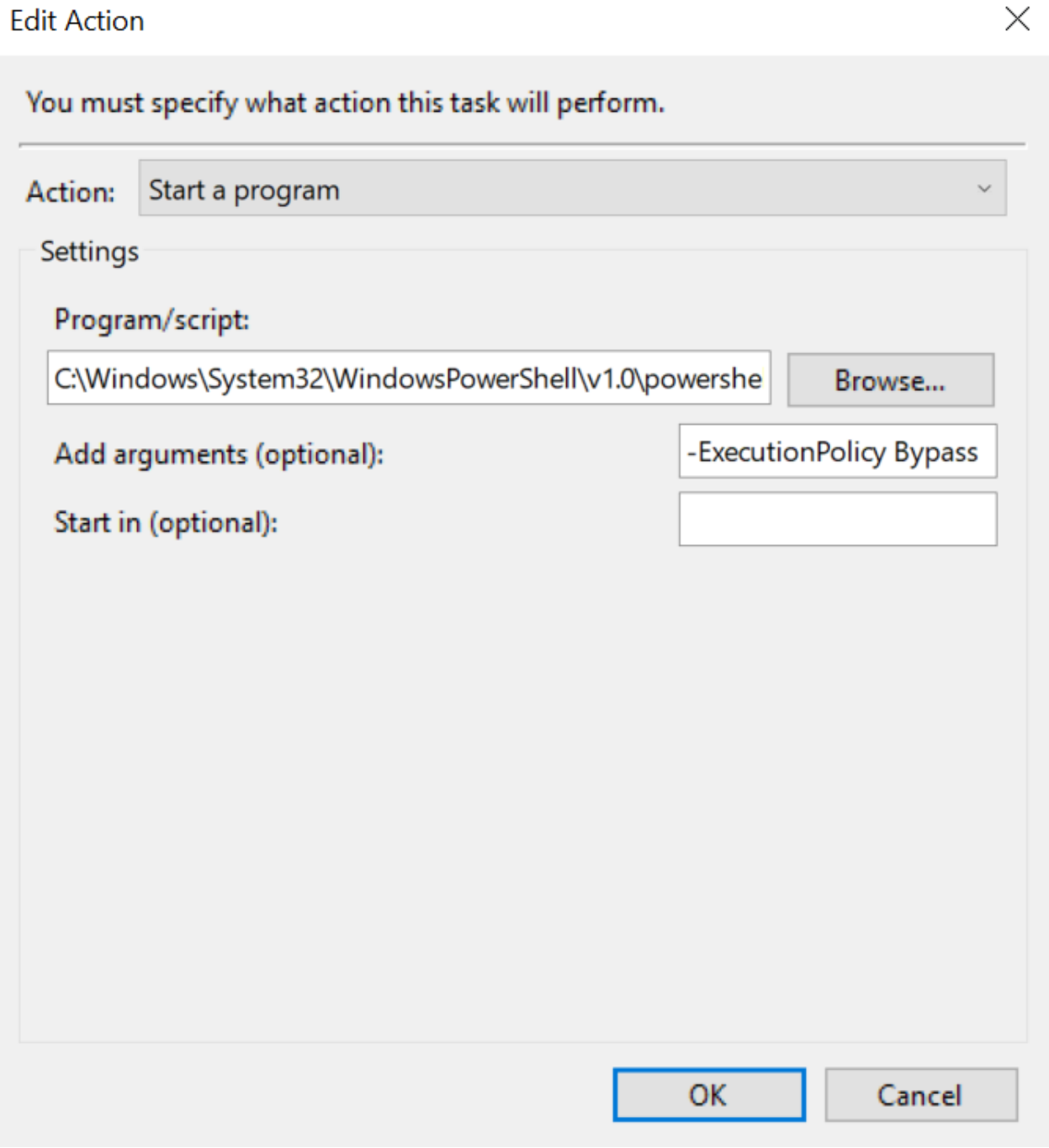


Similar to the previous, I set the triggers for my user either when logging in or unlocking the PC.

Actions



This is more complicated. You must invoke PowerShell and give it permission to run. See below:



The program/script:

C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

The arguments are:

-ExecutionPolicy Bypass -File "C:\Users\kaise\OneDrive\Desktop\AORUSMASTERZ390\MySearch.ps1"

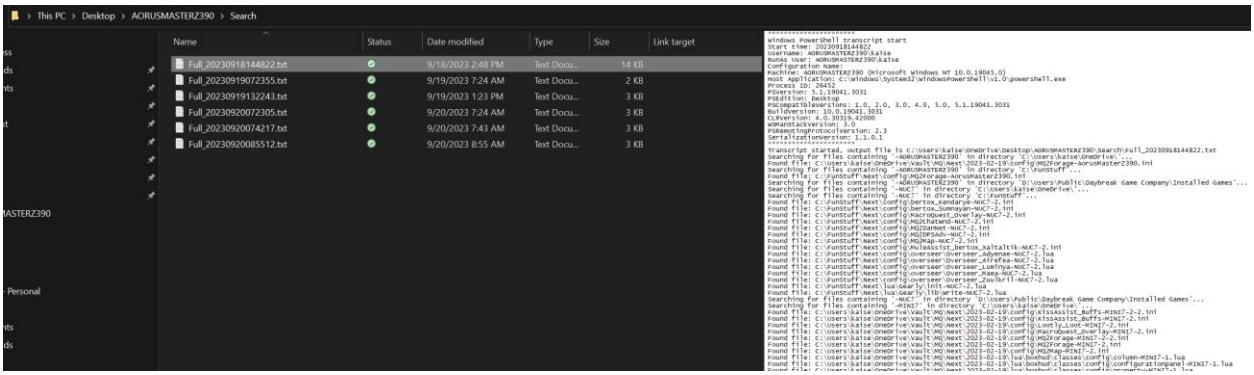
Which tells PowerShell which script to run. In theory, the parameters can be passed in, but I found Windows mangles them when they contain spaces. So, I simply created different script files with the modified parameters saved.

Remaining Tabs

Customize as desired. I left those at the defaults.

Log Files

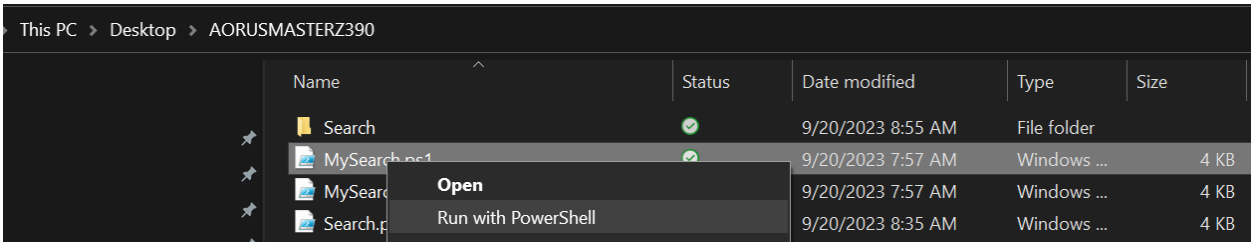
These are stored on your desktop in COMPUTERNAME\Search\ (I place the scripts in the COMPUTERNAME folder for convenience):



In this example, logs with the prefix Full are shown. I only have run MySearch.ps1 on this PC. I use a different log file prefix for each script so that I can retain the history separately.

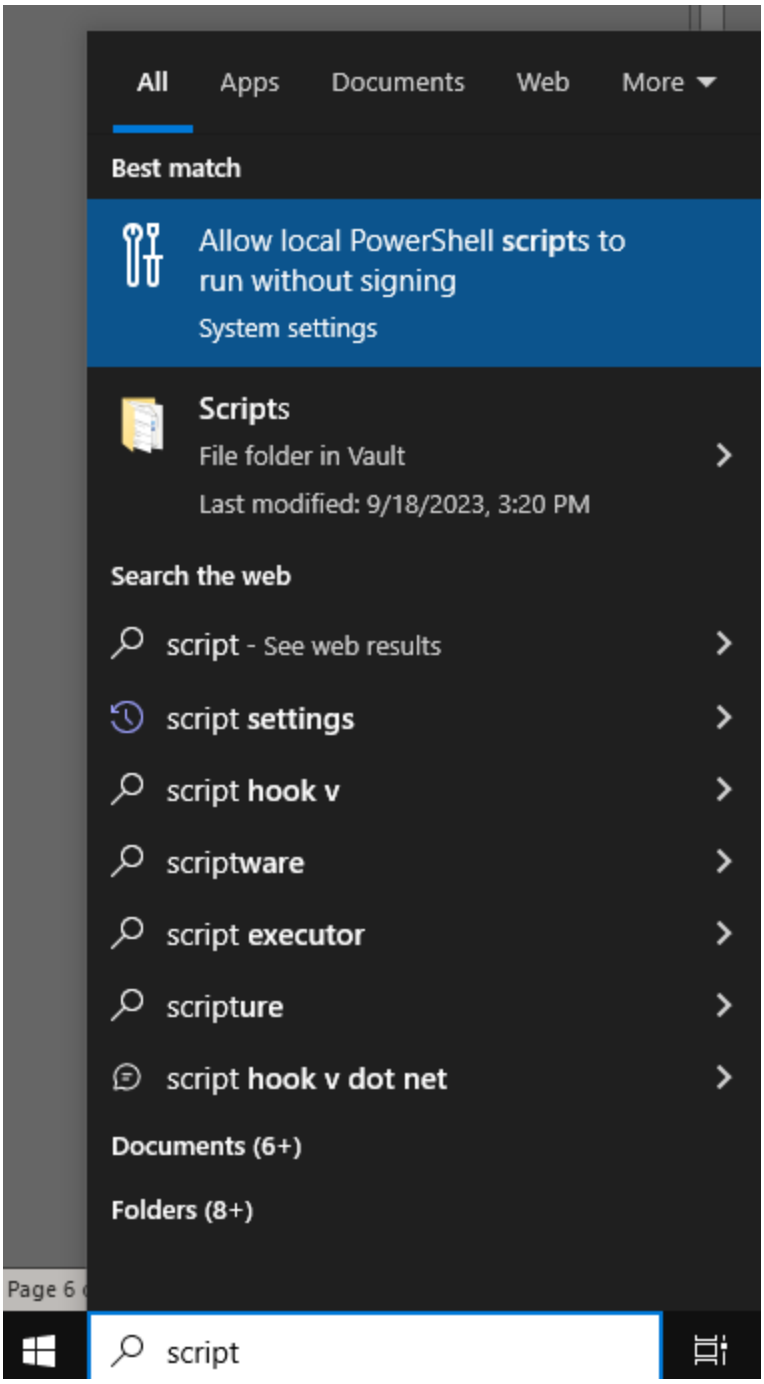
PowerShell Script Execution

When I run the script on demand, I usually right-click and choose Run with PowerShell:

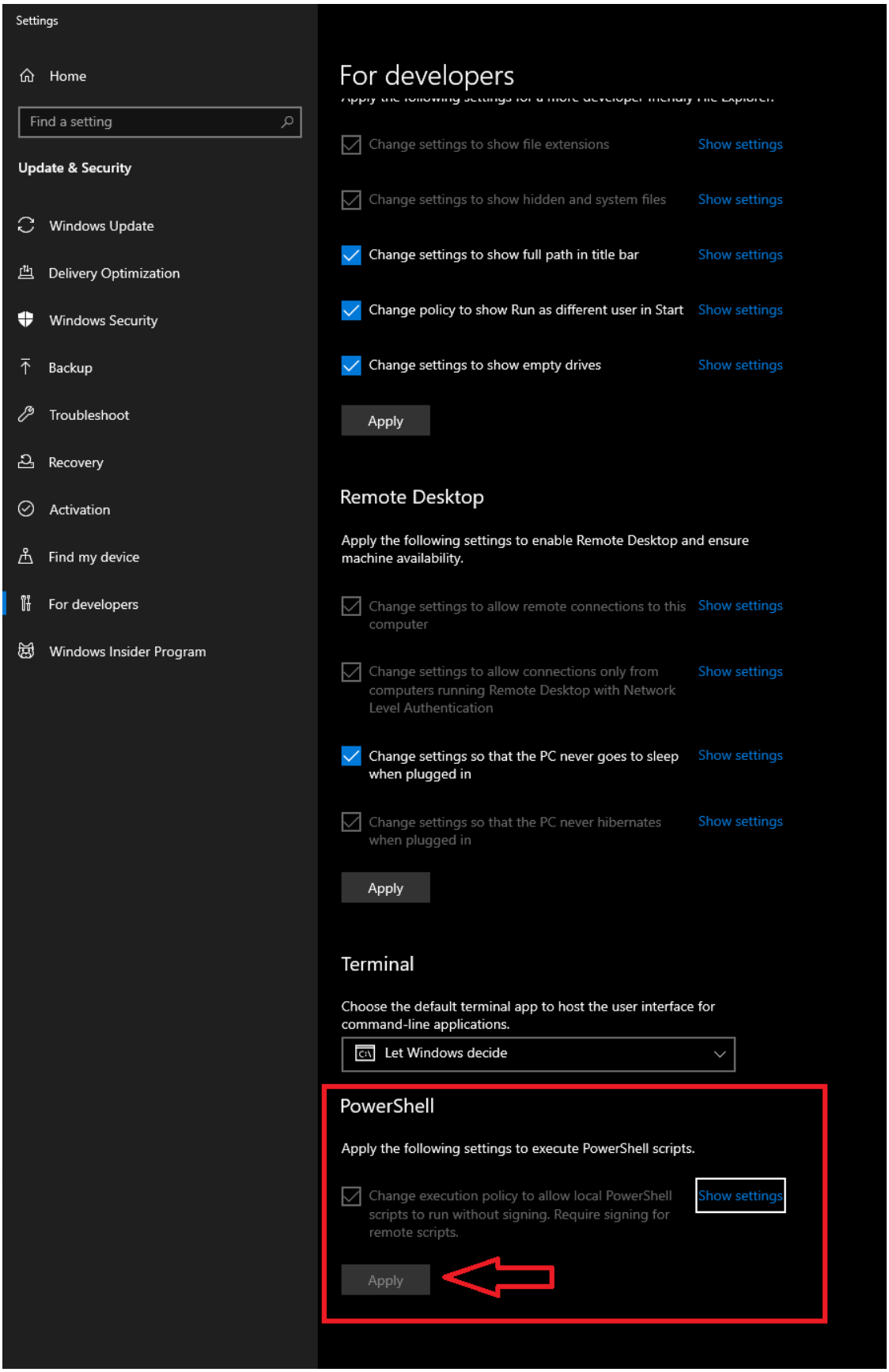


Permission

You should configure Windows to allow PowerShell execution. Win + Q to search for script will find “Allow local PowerShell scripts to run without signing”:



Scroll to the bottom and click the Apply button to allow execution. If it is greyed out, it is already turned on.


Settings
Home
Find a setting
Update & Security
Windows Update
Delivery Optimization
Windows Security
Backup
Troubleshoot
Recovery
Activation
Find my device
For developers
Windows Insider Program
For developers
Change settings to show file extensions
Show settings
Change settings to show hidden and system files
Show settings
Change settings to show full path in title bar
Show settings
Change policy to show Run as different user in Start
Show settings
Change settings to show empty drives
Show settings
Apply
Remote Desktop
Apply the following settings to enable Remote Desktop and ensure machine availability.
Change settings to allow remote connections to this computer
Show settings
Change settings to allow connections only from computers running Remote Desktop with Network Level Authentication
Show settings
Change settings so that the PC never goes to sleep when plugged in
Show settings
Change settings so that the PC never hibernates when plugged in
Show settings
Apply
Terminal
Choose the default terminal app to host the user interface for command-line applications.
Let Windows decide
PowerShell
Apply the following settings to execute PowerShell scripts.
Change execution policy to allow local PowerShell scripts to run without signing. Require signing for remote scripts.
Show settings
Apply