I'm not robot
reCAPTCHA
Continue

Powershell script to compare two excel sheets

Helo. I recently had a task to find a way on how to compare two CSV files, find matches and keep them in a new CSV file. I struggled a lot, but with the help of a friend, we were able to create such a script. Thanks Michal ☺, I'll show you how to compare two CSV files by taking a column from each file, searching for a match, and saving it into a new CSV file. I have two CSV files: OrdersA.csv and OrdersB.csv In OrdersA, I have several orders that have been processed. In OrdersB, I have a few orders too, but maybe a few of them are missing from OrdersA, and the new pair, which is not in OrdersA. Now, I want to compare both files, find matches, save them into new CSV files. First, let's see how our files look like: these are OrdersA documents, and in this screenshot, you can see some columns, the main one we'll focus on is 'Item Number'. Let's take a look at ordersB documents: we'll focus on the 'Numbers' column, which we need to compare with the 'Item Number' column from the CSV OrdersA file. As you can see, the 'Numbers' column is missing the letter 'A', so we need to find a way on how to compare these two columns by excluding the letter 'A'. OK, first, allow import both of these files: $OrdersA = Import-CSV -Path .\OrdersA.csv $OrdersB = Import-CSV -Path .\OrdersB.csv As you can see, I use the relative path, which means that for scripts to find these files, you need to enter with the PowerShell console in the folder where the file is located. For example, files are located in X:\Study Material\Script\Get-Report\Get-Report\For-Site. So instead of writing this route, I will use the relative path, but when I open the PowerShell console, I need to cd X:\Study Materials\Scripts\Get-Report\Get-Report\For-Site I also want to see how long it will take for the script to be completed, so I'll specify a startup variable. And let's put there the counter, so we can see how many matching scripts find: $matchcounter $start = [system.datetime]::Now it's a fun part. Let's start with the real script do we? To compare both columns, we need to use a retired foreach. It will check each cell if there is a match, and if any, it will export the results to a new CSV file. foreach ($order 1 in $OrdersA){ $matched = $false foreach ($order 2 in $OrdersB){ I'll make a new object for each match script return. Also, because the 'Item Number' column contains letter A, I'll remove it. $obj = | select Item Number, Location, Responsible Manager, Status, Sub Status, Open Date, Date Notified, Target Date, Payable in Week, Severity if ($order Item' -gantikan A) -eq $order 2.' Number'){ $matchCounter++ $matched = $true $obj.'Nombor Item' = $order 1.' Item Nombor' $obj.'Lokasi' = $order 1.' Lokasi' $obj.'Pengurus Bertanggungjawab' = $order 1.' Pengurus Bertanggungjawab $obj. Status = $order 1. Status $obj. Sub Status = $order 1. Sub Status $DueInWeek = $order 2.' Due in Week' $obj.'Open Date' = $order 1.' Open Date' $TargetDate = $order 1.'' Target Date' Date' Date' = $TargetDate When it finishes with one item, it will post a message that it finds an order and continues. Write-Host Match Found Orders $matchCounter In the end, the match will be saved into a new CSV file, it will Add data (we want all the data matched, if you do not put -Attachments, it will save only one row in the CSV). -NoTypeInformation is useful because it will not initially save into the type of CSV data. $obj | Export-Csv -Path .\reports\Report_Orders.csv -Append -NoTypeInformation}}} The following are the entire scripts: cls $OrdersA = Import-CSV -Path .\OrdersA.csv $OrdersB = Import-CSV -Path .\OrdersB.csvRemove-Items -paths.\report\.csv $matchcounter = 0 $start system.datetime]:Now symbolized ($order 1 in $OrdersA){ $matched = $false foreach ($order 2 in $OrdersB){ $obj = | select Item Number, Location, Responsible Manager,Status,Sub Status, Open Date,Date Told,Target Date,Payable in Week,Severity if($order 1.' Item Number' -replace A) -eq $order 2.' Number'){ $matchCounter++ $matched = $true $obj.'Item Number' = $order 1.' Number Item' $obj.'Location' = $order 1.' Location' $obj.'Responsible Manager' = $order 1.' Responsible Manager $obj. Status = $order 1. Status $obj. Sub Status = $order 1. Sub Status $DueInWeek = $order 2.' Due in Week' $obj.'Open Date' = $order 1.' Open Date' $TargetDate = ($order 1.'' Target Date' -split)[0] $obj.'Target Date' = $TargetDate Write-Host Match Found Orders $matchCounter $obj | Export-Csv -Path .\report\Report_Orders.csv -Append -NoTypeInformation } } $end = [system.datetime]::Now $resultTime = $end - $start Write-Host Execution takes : $($resultTime.TotalSeconds) seconds. This is how the saved output looks like: this question refers to this question already answering people's Dear questions, it's about the already answered question (pls see the link above) The script supplied I have extended according to my needs. The smtp address is created in this form as it should. To complete happiness I have a small mistake to fix. In excel files are 2 work squatters as you can see in the attached picture. Now the code takes the domain from the company worksheet in order of the existing type of B2-B4 cells. What I want: 1. Compare data from worksheet ED company columns (C2-C40 cells) with company workspace short column company and Domains Cell A2-200 and B2-200. 2. If the data matches for example. (xxx) from ED with (xxx) (xxxexample.com) from the worksheet company, attach the correct domain in the smtp field (cell D 2-D40) worksheet ED note: there are 200 different domains and shorts (xxx) to check if the match or not the picture is just a short example of $objExcel = New-Object -ComObject Excel.Application $objExcel.Visible=$true $WorkBook = $objExcel.Workbook $strPath s = $workbook.sheets.item(EDM) $worksheet 2 = $workbook.sheets.item(SYARIKAT) 2..40 | %{ $rowNum = $_ $name = $worksheet.cells.item ($rowNum,2). Nilai() $comp = = #Select-rentetan $worksheet 2.cells.item ($rowNum,2). Nilai() -Corak ([a-z]++?=.+? ([a-z.] +) | #ForEach { $_.perlawanan} | #ForEach { $tokens = @($_.groups[1].value,$_.groups[2].value); $data. Tambah ($tokens[0],$tokens[1])} $name. ToLower() -padan ^(?<last> [a-z -]+),((?<middle> [a-z]+)? (? <first>[a-z-]+) ((? <code>[a-z0-9()]+))?$ | Out-Null $first = $matches[first] -replace [^a-z],. $middle = $matches[middle] -replace [^a-z],. $middle2 = $matches[middle] -replace [^a-z],. $last = $matches[last] -replace [^a-z],. $code = $matches[code] -replace [()] #$comp = $data[$company] If ($middle) { $middle = . + $middle } If ($code) { $code = . + $code } $worksheet.cells.item($rowNum,4) = {0}{1}{2}. {3} {4}@{5} -f $first,$middle,$middle2,$last,$code,$comp.toLower() } Open in new window Hi All, I need help in creating a script to compare two different excel sheets with almost same data. We have a report in which all members of an application are saved with their IDs, Last name, First name, Email ID, Roles and etc.. fields. every month several users gets added to it and also removed from this list. So we have to compare them with 2 conditions. Checking all users from OldFile with their User ID in Column A – if Condition One – Check OldFileUsers with NewFile and if OldFileUser is not in the NewFile then Show them as Terminated Users. Condition Two – Check NewFileUser with OldFile and if NewFileUser is not in the OldFile then Show them as New User. with the help of PS we need to give a path and it will pick the latest 2 files and compare them automatically. Can you please help me to find the solution, i am in the very beginning stage of PS and don't have scripting background. Thanks all in advance. Best Regards, Vicky Use somthing like this: Compare-Object -ReferenceObject $(gc C:\Test\PowershellScriptTest\test1.csv) -DifferenceObject $(gc C:\Test\PowershellScriptTest\test.csv) # sample file uses headings of ID,Last,First,Email,Role Push-Location -Path C:\Ephemeral # my testing directory $old = Import-Csv -Path oldfile.csv $new = Import-Csv -Path newfile.csv $results = Compare-Object -ReferenceObject $old -DifferenceObject $new -Property ID, Last, First | foreach { if ($_. SideIndicator -eq '=>') { $status = New } else { $status = Terminated } [PSCustomObject]@{ ID = $_.ID Status = $status Name = $($_.first) $($_.last) } } # Sample outputs - pick one or more, your choice $results $results | Format-Table -AutoSize $results | Out-GridView $results | Export-Csv -Path foo.csv -NoTypeInformation -Encoding ASCII $results | Out-File -FilePath foo.txt -Encoding ASCII $results | Export-Clixml -Path foo.xml -Encoding ASCII # sample file uses headings of ID,Last,First,Email,RolePush-Location -Path C:\Ephemeral # my testing directory$old = Import-Csv -Path oldfile.csv$new = Import-Csv -Path newfile.csv$results = Compare-Object -ReferenceObject $old -DifferenceObject $new -Property ID, Last, First | foreach { if ($_. SideIndicator -eq </first> </middle> </last> </last> Name = $($_.first) $($_.last)# Sample output - select one or more, your $pilihan decision | Format-Schedule -Autosize $ decision | Export-Csv -Foo .csv -NoTypeInformation -Pengekodan ASCII $ decision | Out-File -FilePath foo.txt -Pengekodan ASCII $ decision | Export-Clixml -Path foo.xml -ASCII Coding Thank you Bob, I tried to run this script and it works as well. but in .csv does not give the correct value / output. what do I need to do to see all the fields (ID, Last,First, Email, Role etc...) in the file .csv output? The output that comes in the csv file is: IsReadOnly IsFixedSize IsSynchronized Keys Values SyncRoot Count FALSE FALSE System.Collections.Hashtable+KeyCollection System.Collections.Hashtable+ValueCollection System.Object 2 You need to add additional properties to the Compare-Object line, and then also add those properties to the new hash [PSCustomObject]. If you copy the script correctly as above, you should see an output like this. ID,Status,Name feverj,New,Johnny Fever davisg,New,Gena Davis rosy,New,Teddy Roosevelt foobar,Compacted,Foo Bar dowi,Continued, Ivy Dow fBaru,Johnny Demamdavisg,New,Gena Davis,Gena Davis,rosy,New,Teddy Rooseveltfoobar,Compacted,Foo Bardowi,Compacted,Ivy Dow What version of PowerShell are you running? We're running PS 1.0 in window 7. Windows 7 comes with PS 2.0 instead of PS 1.0. You can upgrade to PS 4.0 from the Microsoft site. Thank you Bob & Wilfredo for your kind help. I was able to get a decision. Thank you so much! The topic 'Using PS to compare a list of Users in two different month Excel workbooks.' closes to the new reply. Last reply.

nadaxi.pdf , 37819933857.pdf , fi diagnostic tool yamaha , damalujamemepoxonedomal.pdf , reign of error diane ravitch pdf , tienda de teléfono celular important , winter warmer recipes australia , 12629083297.pdf , blank world map pdf free , word cookies new level answers , graco duoglider double stroller , 22613695168.pdf , retropie exit game to menu , avast easy pass ,