

How to combine multiple CSVs using PSCustomObject to merge/append data sets.

By CountDClemo

To test the script, place the following 3 csv files in a Desktop folder named “test-custom-object”.

 Name-Phone-Stnumber-Stname stnumber-stname-datepurchased stnumber-stname-homeval-tax

Goal:

To combine multiple CSVs at least one CSV requires a data set that is common and/or made common to the CSVs to be combined. In this test case there are two data sets that are common to all which are Street number and Street Name.

Best to utilize a base data set that has all required reportable attributes.

For example, In the base data set (Name-Phone-Stnumber-Stname.csv) there is a row listed as “Test, Person” which is not present in the other CSVs; however, in the case of this report we will want “Test, Person” to be listed even if he is not present in the other CSVs.

Test,Person	555-000	10000	NoCsv RD	
-------------	---------	-------	----------	--

Name-Phone-Streetnumber-Streetname.csv will be the primary csv that we use to compare and appended matching data of the other CSVs.

Also, we have street name data sets that are the same; however, they are filtered with where-object cmdlet that must match both the street name and street number.

```
where-object {$_.'Street Name' -eq $Streetname -and $_.'Street number' -eq $Streetnumber}
```

The 3 CSVs and their expected output:

Name-Phone-Stnumber-Stname.csv

Name	Phone	Street number	Street Name
Bill,Salman	555-0986	15009	Holbert Rd
June,Newman	555-0997	23887	Towns Rd
Sandy, Burke	555-7642	34562	Ownes Ave
Jason, Paul	555-1234	97654	Hiltop Ln
Jim, Herndandez	555-4321	87632	Archer Rd
Phil, Goldman	555-9876	15678	Samsonview Rd
Joe, Finders	555-9832	15003	Holbert Rd
Gabe, Carpenter	555-0091	13807	Towns Rd
Julie, Sanders	555-8824	20562	Ownes Ave
Kim, Blacksmith	555-0067	77602	Archer Rd
Test,Person	555-000	10000	NoCsv RD

stnumber-stname-datepurchased.csv

Street number	Street Name	Date Purchases
15009	Holbert Rd	9/8/1999
23887	Towns Rd	9/4/1997
34562	Ownes Ave	1/5/2001
97654	Hiltop Ln	3/11/2000
87632	Archer Rd	7/9/2005
15678	Samsonview Rd	28/2/2009
15003	Holbert Rd	15/7/1998
13807	Towns Rd	3/12/1990
20562	Ownes Ave	2/10/2018
77602	Archer Rd	3/3/2003

stnumber-stname-homeval-tax.csv

Street number	Street Name	Home Value	Tax
15009	Holbert Rd	125000	3750
23887	Towns Rd	124000	3720
34562	Ownes Ave	186000	5580
97654	Hiltop Ln	156700	4701
87632	Archer Rd	156000	4680
15678	Samsonview Rd	110000	3300
15003	Holbert Rd	150230	4506
13807	Towns Rd	129000	3870
20562	Ownes Ave	203000	6090
77602	Archer Rd	146030	4381

Final-Combination.csv

Name	Phone	Street number	Street Name	Date Purchased	Home Value	Tax
Bill,Salman	555-0986	15009	Holbert Rd	9/8/1999	125000	3750
June,Newman	555-0997	23887	Towns Rd	9/4/1997	124000	3720
Sandy, Burke	555-7642	34562	Ownes Ave	1/5/2001	186000	5580
Jason, Paul	555-1234	97654	Hiltop Ln	3/11/2000	156700	4701
Jim, Herndandez	555-4321	87632	Archer Rd	7/9/2005	156000	4680
Phil, Goldman	555-9876	15678	Samsonview Rd	28/2/2009	110000	3300
Joe, Finders	555-9832	15003	Holbert Rd	15/7/1998	150230	4506
Gabe, Carpenter	555-0091	13807	Towns Rd	3/12/1990	129000	3870
Julie, Sanders	555-8824	20562	Ownes Ave	2/10/2018	203000	6090
Kim, Blacksmith	555-0067	77602	Archer Rd	3/3/2003	146030	4381
Test,Person	555-000	10000	NoCsv RD			

Script explained lines 1-4:

Lines 1 to 4 set import the primary csv and set the variables to be used in the [PSCustomObject] . In this case we have to preset the variables of street name and street number of the primary CSV to avoid any where-object conflicts of other CSV using the same naming conventions.

```
1 Import-CSV "$home\Desktop\test-custom-object\Name-Phone-Stnumber-Stname.csv" | ForEach-Object {  
2  
3     $Streetname = $_.'Street Name'  
4     $Streetnumber = $_.'Street number'  
5 }
```

Script explained lines 5-15:

```
5  
6 [PSCustomObject]@{  
7  
8     Name = $_.name  
9  
10    Phone = $_.Phone  
11  
12    'Street number' = $_.'Street number'  
13  
14    'Street Name' = $_.'Street Name'  
15 }
```

This sets the first four columns of the CSV to that of the primary imported CSV. Note any headers that contain a space will need to be placed in single quotes .ie 'Street number'. If desired the prefix can be changed to what suits your needs for example 'Street number' = \$_.'Street number' can be changed to 'Number of Street' = \$_.'Street number' .

Script explained lines 15-29:

```
15  
16     'Date Purchased' = (import-csv "$home\Desktop\test-custom-object\stnumber-stname-datepurchased.csv" |  
17     where-object {$_. 'Street Name' -eq $Streetname -and $_. 'Street number' -eq $Streetnumber}). 'Date Purchased'  
18  
19     'Home Value' = (import-csv "$home\Desktop\test-custom-object\stnumber-stname-homeval-tax.csv" |  
20     where-object {$_. 'Street Name' -eq $Streetname -and $_. 'Street number' -eq $Streetnumber}). 'Home Value'  
21  
22     Tax = (import-csv "$home\Desktop\test-custom-object\stnumber-stname-homeval-tax.csv" |  
23     where-object {$_. 'Street Name' -eq $Streetname -and $_. 'Street number' -eq $Streetnumber}). Tax  
24  
25  
26 }  
27 } |  
28  
29 Export-Csv -Path "$home\Desktop\test-custom-object\Final-Combination.csv" -NoTypeInfo -Append
```

Lines 16 to 17 import the “stnumber-stname-datepurchased.csv” compares the street name and street number to that of “Name-Phone-Stnumber-Stname.csv” and appends the Date Purchased data to the final product output “Final-Combination.csv”

Likewise Home Value and tax are processed in the same scenario by importing “stnumber-stname-homeval-tax.csv and filtering” it by street number and street name found in “Name-Phone-Stnumber-Stname.csv”

This script can be leveraged and adjusted for data sets containing a few thousand object; however, please keep in mind when dealing with larger data sets it can take four or more seconds per row to find and append output. Therefore, when comparing a 20K plus data set it could take 80,000 seconds or approx. 22hrs to complete. In such a case it would be better to use excel with iferror vlookup formulas.