**Sorting Objects**

* Article
* 10/08/2021
* 2 minutes to read
* 1 contributor

We can organize displayed data to make it easier to scan by using the Sort-Object cmdlet. Sort-Object takes the name of one or more properties to sort on, and returns data sorted by the values of those properties.

**Basic sorting**

Consider the problem of listing subdirectories and files in the current directory. If we want to sort by **LastWriteTime** and then by **Name**, we can do it by typing:

PowerShellCopy

Get-ChildItem |

Sort-Object -Property LastWriteTime, Name |

Format-Table -Property LastWriteTime, Name

OutputCopy

LastWriteTime Name

------------- ----

11/6/2017 10:10:11 AM .localization-config

11/6/2017 10:10:11 AM .openpublishing.build.ps1

11/6/2017 10:10:11 AM appveyor.yml

11/6/2017 10:10:11 AM LICENSE

11/6/2017 10:10:11 AM LICENSE-CODE

11/6/2017 10:10:11 AM ThirdPartyNotices

11/6/2017 10:10:15 AM tests

6/6/2018 7:58:59 PM CONTRIBUTING.md

6/6/2018 7:58:59 PM README.md

...

You can also sort the objects in reverse order by specifying the **Descending** switch parameter.

PowerShellCopy

Get-ChildItem |

Sort-Object -Property LastWriteTime, Name -Descending |

Format-Table -Property LastWriteTime, Name

OutputCopy

LastWriteTime Name

------------- ----

12/1/2018 10:13:50 PM reference

12/1/2018 10:13:50 PM dsc

...

6/6/2018 7:58:59 PM README.md

6/6/2018 7:58:59 PM CONTRIBUTING.md

11/6/2017 10:10:15 AM tests

11/6/2017 10:10:11 AM ThirdPartyNotices

11/6/2017 10:10:11 AM LICENSE-CODE

11/6/2017 10:10:11 AM LICENSE

11/6/2017 10:10:11 AM appveyor.yml

11/6/2017 10:10:11 AM .openpublishing.build.ps1

11/6/2017 10:10:11 AM .localization-config

**Using hash tables**

You can sort different properties in different orders by using hash tables in an array. Each hash table uses an **Expression** key to specify the property name as string and an **Ascending** or **Descending** key to specify the sort order by $true or $false. The **Expression** key is mandatory. The **Ascending** or **Descending** key is optional.

The following example sorts objects in descending **LastWriteTime** order and ascending **Name** order.

PowerShellCopy

Get-ChildItem |

Sort-Object -Property @{ Expression = 'LastWriteTime'; Descending = $true },

@{ Expression = 'Name'; Ascending = $true } |

Format-Table -Property LastWriteTime, Name

OutputCopy

LastWriteTime Name

------------- ----

12/1/2018 10:13:50 PM dsc

12/1/2018 10:13:50 PM reference

11/29/2018 6:56:01 PM .openpublishing.redirection.json

11/29/2018 6:56:01 PM gallery

11/24/2018 10:33:22 AM developer

11/20/2018 7:22:19 PM .markdownlint.json

...

You can also set a scriptblock to the **Expression** key. When running the Sort-Object cmdlet, the scriptblock is executed and the result is used for sorting.

The following example sorts objects in descending order by the time span between **CreationTime** and **LastWriteTime**.

PowerShellCopy

Get-ChildItem |

Sort-Object -Property @{ Expression = { $\_.LastWriteTime - $\_.CreationTime }; Descending = $true } |

Format-Table -Property LastWriteTime, CreationTime

OutputCopy

LastWriteTime CreationTime

------------- ------------

12/1/2018 10:13:50 PM 11/6/2017 10:10:11 AM

12/1/2018 10:13:50 PM 11/6/2017 10:10:11 AM

11/7/2018 6:52:24 PM 11/6/2017 10:10:11 AM

11/7/2018 6:52:24 PM 11/6/2017 10:10:15 AM

11/3/2018 9:58:17 AM 11/6/2017 10:10:11 AM

10/26/2018 4:50:21 PM 11/6/2017 10:10:11 AM

11/17/2018 1:10:57 PM 11/29/2017 5:48:30 PM

11/12/2018 6:29:53 PM 12/7/2017 7:57:07 PM

...

**Tips**

You can omit the **Property** parameter name as following:

PowerShellCopy

Sort-Object LastWriteTime, Name

Besides, you can refer to Sort-Object by its built-in alias, sort:

PowerShellCopy

sort LastWriteTime, Name

The keys in the hash tables for sorting can be abbreviated as following:

PowerShellCopy

Sort-Object @{ e = 'LastWriteTime'; d = $true }, @{ e = 'Name'; a = $true }

In this example, the **e** stands for **Expression**, the **d** stands for **Descending**, and the **a** stands for **Ascending**.

To improve readability, you can place the hash tables into a separate variable:

PowerShellCopy

$order = @(

@{ Expression = 'LastWriteTime'; Descending = $true }

@{ Expression = 'Name'; Ascending = $true }

)

Get-ChildItem |

Sort-Object $order |

Format-Table LastWriteTime, Name