SS64 | PowerShell > | How-to > | Search

ForEach-Ob

Perform an operati pipeline.

(SS64)

We value your privacy

We and our partners store and/or access information on a device, such as cookies and process personal data, such as unique identifiers and standard information sent by a device for personalised advertising and content, advertising and content measurement, audience research and services development. With your permission we and our partners may use precise geolocation data and identification through device scanning. You may click to consent to our and our 1414 partners' processing as described above. Alternatively you may click to refuse to consent or access more detailed information and change your preferences before consenting. Please note that some processing of your personal data may not require your consent, but you have a right to object to such processing. Your preferences will apply to this website only. You can change your preferences or withdraw your consent at any time by returning to this site and clicking the "Privacy" button at the bottom of the webpage. Please note that this website/app uses one or more Google services and may gather and store information including but not limited to your visit or usage behaviour. You may click to grant or deny consent to Google and its third-party tags to use your data for below specified purposes in below Google consent section.

MORE OPTIONS | DISAGREE | AGREE

```
Syntax
      ForEach-Ob
         [-begin

key
   -begin Scri
       A scrip

   -end Script
       A scrip

   -process Sc
       A block

   -inputObjec
       Accepts
       Enter a                                                ject(s).

       When th
       ForEach
       Because                                                anged.

       To proc

       This pa                                                and its
       direct

   -Parallel
       Run all script blocks in parallel for each piped input object. PowerShell 7.0+
       Parallel will not always speed up script execution. And can significantly slow down script
       execution if used heedlessly. The overhead for a trivial script can make -parallel much slower.

   -ThrottleLimit
       Limit the number of script blocks running in parallel at a time, default = 5. PowerShell 7.0+
```

Standard Aliases for Foreach-Object: the '`%`' symbol, `ForEach`

For operations in the pipeline, the `ForEach` alias will take precedence over the ForEach statement.
For operations not in the pipeline the ForEach statement will take precedence.

For the fastest performance: use the ForEach statement (or method) when the collection of objects is small enough that it can be loaded into memory. (eg an array of 20 string values)
Use the `ForEach-Object` cmdlet when you want to pass only one object at a time through the pipeline, minimising memory usage. (e.g. a directory containing 10,000 files)

Examples

Retrieve the files (and folders) from the C: drive and display the size of each:

```
PS C:> Get-ChildItem C:\ | ForEach-Object -process { $_.length / 1024 }
```

(The $_ variable holds a reference to the current item being processed.)

Retrieve the 500 most recent events from the system event log and store them in the $events variable:

```
PS C:> $events = Get-Eventlog -logname system -newest 500
```

Then pipe the $events variable into the ForEach-Object cmdlet.

```
$events | ForEach-Object -begin {
   Get-Date
} -process {
   Out-File -filepath event_log.txt -append -inputobject $_.message
} -end {
   Get-Date
}
```

In this example, the -Process parameter uses Out-File to create a text file and stores the .message property of each event.
The -`Begin` and -End parameters are used to display the date/time before and after the `-process` command is run.

"The best way to have a good idea is to have a lot of ideas" ~ Linus Pauling

Related PowerShell Cmdlets

ForEach statement.
ForEach (method) - Loop through items in a collection.
Compare-Object C
Group-Object - Gro
Invoke-Parallel - via lel).
Measure-Object - M
New-Object - Crea
Select-Object - Se
Sort-Object - Sort t
Tee-Object - Send
Where-Object - Fil