Full credit is given to the above companies including the OS that this PDF file was generated!

PowerShell Get-Help on command 'Get-Unique'

PS C:\Users\wahid> Get-Help Get-Unique

```
NAME
    Get-Unique

SYNOPSIS
    Returns unique items from a sorted list.


SYNTAX
    Get-Unique [-AsString] [-InputObject <System.Management.Automation.PSObject>]
    [<CommonParameters>]

    Get-Unique [-InputObject <System.Management.Automation.PSObject>] [-OnType]
    [<CommonParameters>]


DESCRIPTION
    The `Get-Unique` cmdlet compares each item in a sorted list to the next item,
    eliminates duplicates, and returns only one instance of each item. The list
    must be sorted for the cmdlet to work properly.

    `Get-Unique` is case-sensitive. As a result, strings that differ only in
```

character casing are considered to be unique.

PARAMETERS

-AsString <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet uses the data as a string. Without this parameter, data is treated as an object, so when you submit a collection of objects of the same type to `Get-Unique`, such as a collection of files, it returns just one (the first). You can use this parameter to find the unique values of object properties, such as the file names.

-InputObject <System.Management.Automation.PSObject>

Specifies input for `Get-Unique`. Enter a variable that contains the objects or type a command or expression that gets the objects.

This cmdlet treats the input submitted by using InputObject as a collection. it does not enumerate individual items in the collection. Because the collection is a single item, input submitted by using InputObject is always returned unchanged.

-OnType <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet returns only one object of each type.

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (https:/go.microsoft.com/fwlink/?LinkID=113216).

---------- Example 1: Get unique words in a text file ----------

```
$A = $( foreach ($line in Get-Content C:\Test1\File1.txt) {
    $line.tolower().split(" ")
```

```
  }) | Sort-Object | Get-Unique
$A.count
```

The first command gets the content of the `File.txt` file. It converts each line of text to lowercase letters and then splits each word onto a separate line at the space (`" "`). Then, it sorts the resulting list alphabetically (the default) and uses the `Get-Unique` cmdlet to eliminate any duplicate words. The results are stored in the `$A` variable.

The second command uses the Count property of the collection of strings in `$A` to determine how many items are in `$A`.

---------- Example 2: Get unique integers in an array ----------

```
1,1,1,1,12,23,4,5,4643,5,3,3,3,3,3,3,3 | Sort-Object | Get-Unique

1
3
4
5
12
23
4643
```

The first command takes an array of integers typed at the command line, pipes them to the `Sort-Object` cmdlet to be sorted, and then pipes them to `Get-Unique`, which eliminates duplicate entries.

------ Example 3: Get unique object types in a directory ------

```
Get-ChildItem | Sort-Object {$_.GetType()} | Get-Unique -OnType
```

The pipeline operator (`|`) sends the results to the `Sort-Object` cmdlet. The `$_.GetType()` statement applies the GetType method to each file or directory. Then, `Sort-Object` sorts the items by type. Another pipeline operator sends the results to `Get-Unique`. The OnType parameter directs `Get-Unique` to return only one object of each type.

-------------- Example 4: Get unique processes --------------

```
Get-Process | Sort-Object | Select-Object processname | Get-Unique -AsString
```

The `Get-Process` command gets all of the processes on the computer. The pipeline operator (`|`) passes the result to `Sort-Object`, which, by default, sorts the processes alphabetically by ProcessName . The results are piped to the `Select-Object` cmdlet, which selects only the values of the ProcessName property of each object. The results are then piped to `Get-Unique` to eliminate duplicates.

The AsString parameter tells `Get-Unique` to treat the ProcessName values as strings. Without this parameter, `Get-Unique` treats the ProcessName values as objects and returns only one instance of the object, that is, the first process name in the list.

REMARKS

To see the examples, type: "get-help Get-Unique -examples".

For more information, type: "get-help Get-Unique -detailed".

For technical information, type: "get-help Get-Unique -full".

For online help, type: "get-help Get-Unique -online"