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

PowerShell Get-Help on command 'Get-Date'

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

NAME

Get-Date

SYNOPSIS

Gets the current date and time.

SYNTAX

Get-Date [[-Date] <System.DateTime>] [-Day <System.Int32>] [-DisplayHint {Date | Time | DateTime}] [-Format <System.String>] [-Hour <System.Int32>] [-Millisecond <System.Int32>] [-Minute <System.Int32>] [-Month <System.Int32>] [-Second <System.Int32>] [-Year <System.Int32>] [<CommonParameters>]

Get-Date [[-Date] <System.DateTime>] [-Day <System.Int32>] [-DisplayHint {Date | Time | DateTime}] [-Hour <System.Int32>] [-Millisecond <System.Int32>] [-Minute <System.Int32>] [-Month <System.Int32>] [-Second <System.Int32>] [-UFormat <System.String>] [-Year <System.Int32>] [<CommonParameters>]

DESCRIPTION

The `Get-Date` cmdlet gets a DateTime object that represents the current date

or a date that you specify. `Get-Date` can format the date and time in several .NET and UNIX formats. You can use `Get-Date` to generate a date or time character string, and then send the string to other cmdlets or programs.

`Get-Date` uses the current culture settings of the operating system to determine how the output is formatted. To view your computer's settings, use `(Get-Culture).DateTimeFormat`.

PARAMETERS

-Date <System.DateTime>

Specifies a date and time. Time is optional and if not specified, returns 00:00:00. Enter the date and time in a format that is standard for the currently selected locale. You can change the current locale using the `Set-Culture` cmdlet.

For example, in US English:

`Get-Date -Date "6/25/2019 12:30:22"` returns Tuesday, June 25, 2019 12:30:22

-Day <System.Int32>

Specifies the day of the month that is displayed. Enter a value from 1 to 31.

If the specified value is greater than the number of days in a month, PowerShell adds the number of days to the month. For example, `Get-Date -Month 4 -Day 31` displays May 1 , not April 31 .

-DisplayHint <Microsoft.PowerShell.Commands.DisplayHintType>

Determines which elements of the date and time are displayed.

The accepted values are as follows:

- Date : displays only the date - Time : displays only the time - DateTime : displays the date and time

-Format <System.String>

Displays the date and time in the Microsoft .NET Framework format indicated by the format specifier. The Format parameter outputs a String object.

For a list of available .NET format specifiers, see Custom date and time format strings (/dotnet/standard/base-types/custom-date-and-time-format-strings).

When the Format parameter is used, `Get-Date` only gets the DateTime object's properties necessary to display the date. As a result, some of the properties and methods of DateTime objects might not be available.

Starting in PowerShell 5.0, you can use the following additional formats as values for the Format parameter.

- FileDate . A file or path-friendly representation of the current date in local time. The format is `yyyyMMdd` (case-sensitive, using a 4-digit year, 2-digit month, and 2-digit day). For example: 20190627.

- FileDateUniversal . A file or path-friendly representation of the current date in universal time (UTC). The format is `yyyyMMddZ` (case-sensitive, using a 4-digit year, 2-digit month, 2-digit day, and the letter `Z` as the UTC indicator). For example: 20190627Z.

- FileDateTime . A file or path-friendly representation of the current date and time in local time, in 24-hour format. The format is `yyyyMMddTHHmmssffff` (case-sensitive, using a 4-digit year, 2-digit month, 2-digit day, the letter `T` as a time separator, 2-digit hour,

2-digit minute, 2-digit second, and 4-digit millisecond). For example: 20190627T0840107271.

- FileDateTimeUniversal . A file or path-friendly representation of the current date and time in universal time (UTC), in 24-hour format. The format is `yyyyMMddTHHmmssffffZ` (case-sensitive, using a 4-digit year, 2-digit month, 2-digit day, the letter `T` as a time separator, 2-digit hour, 2-digit minute, 2-digit second, 4-digit millisecond, and the letter `Z` as the UTC indicator). For example: 20190627T1540500718Z.

-Hour <System.Int32>

Specifies the hour that is displayed. Enter a value from 0 to 23.

-Millisecond <System.Int32>

Specifies the milliseconds in the date. Enter a value from 0 to 999.

This parameter was introduced in PowerShell 3.0.

-Minute <System.Int32>

Specifies the minute that is displayed. Enter a value from 0 to 59.

-Month <System.Int32>

Specifies the month that is displayed. Enter a value from 1 to 12.

-Second <System.Int32>

Specifies the second that is displayed. Enter a value from 0 to 59.

-UFormat <System.String>

Displays the date and time in UNIX format. The UFormat parameter outputs a string object. UFormat specifiers are preceded by a percent sign (`%`), for example, `%m`, `%d`, and `%Y`. The Notes (#notes)section contains a table of valid UFormat specifiers .

-Year <System.Int32>

Specifies the year that is displayed. Enter a value from 1 to 9999.

<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 the current date and time ----------

Get-Date

Tuesday, June 25, 2019 14:53:32

----- Example 2: Get elements of the current date and time -----

Get-Date -DisplayHint Date

Tuesday, June 25, 2019

`Get-Date` uses the DisplayHint parameter with the Date argument to get only
the date.

Example 3: Get the date and time with a .NET format specifier

Get-Date -Format "dddd MM/dd/yyyy HH:mm K"

Tuesday 06/25/2019 16:17 -07:00

`Get-Date` uses the Format parameter to specify several format specifiers.

The .NET format specifiers used in this example are defined as follows:

| Specifier | Definition | |
--- | --- | | `dddd`
| Day of the week - full name | | `MM` | Month
number | | `dd` | Day of the
month - 2 digits | | `yyyy` | Year in 4-digit
format | | `HH:mm` | Time in 24-hour format -
no seconds | | `K` | Time zone offset from Universal
Time Coordinate (UTC) |

For more information about .NET format specifiers, see Custom date and time
format strings
(/dotnet/standard/base-types/custom-date-and-time-format-strings).
-- Example 4: Get the date and time with a UFormat specifier --

Get-Date -UFormat "%A %m/%d/%Y %R %Z"

Tuesday 06/25/2019 16:19 -07

`Get-Date` uses the UFormat parameter to specify several format specifiers.

The UFormat format specifiers used in this example are defined as follows:

| Specifier | Definition | |
--- | --- | | `%A`
| Day of the week - full name | | `%m` | Month
number | | `%d` | Day of the
month - 2 digits | | `%Y` | Year in 4-digit
format | | `%R` | Time in 24-hour format -
no seconds | | `%Z` | Time zone offset from Universal
Time Coordinate (UTC) |

For a list of valid UFormat format specifiers, see the Notes (#notes)section.

---------- Example 5: Get a date's day of the year ----------

```
(Get-Date -Year 2020 -Month 12 -Day 31).DayOfYear
```

```
366
```

`Get-Date` uses three parameters to specify the date: Year , Month , and Day .
The command is wrapped with parentheses so that the result is evaluated by the
DayofYear property.
Example 6: Check if a date is adjusted for daylight saving time

```
$DST = Get-Date
$DST.IsDaylightSavingTime()
```

```
True
```

A variable, `$DST` stores the result of `Get-Date`. `$DST` uses the
IsDaylightSavingTime method to test if the date is adjusted for daylight
saving time.
------- Example 7: Convert the current time to UTC time -------

```
Get-Date -UFormat "%A %B/%d/%Y %T %Z"
$Time = Get-Date
$Time.ToUniversalTime()
```

```
Wednesday June/26/2019 10:45:26 -07
```

```
Wednesday, June 26, 2019 17:45:26
```

`Get-Date` uses the UFormat parameter with format specifiers to display the
current system date and time. The format specifier %Z represents the UTC
offset of -07 .

The `$Time` variable stores the current system date and time. `$Time` uses the `ToUniversalTime()` method to convert the time based on the computer's UTC offset.

---------- Example 8: Create a timestamp ----------

```
$timestamp = Get-Date -Format o | ForEach-Object { $_ -replace ":", "." }
New-Item -Path C:\Test\$timestamp -Type Directory

Directory: C:\Test

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         6/27/2019    07:59
2019-06-27T07.59.24.4603750-07.00
```

The `$timestamp` variable stores the results of a `Get-Date` command. `Get-Date` uses the Format parameter with the format specifier of lowercase `o` to create a timestamp String object. The object is sent down the pipeline to `ForEach-Object`. A ScriptBlock contains the `$_` variable that represents the current pipeline object. The timestamp string is delimited by colons that are replaced by periods.

`New-Item` uses the Path parameter to specify the location for a new directory. The path includes the `$timestamp` variable as the directory name. The Type parameter specifies that a directory is created.

---------- Example 9: Show invariant culture ----------

```
# Get date using current culture en-US
(Get-Date 2024-03-19).ToString()

3/19/2024 12:00:00 AM

# Get date using invariant culture
```

"$(Get-Date 2024-03-19)"

03/19/2024 00:00:00

REMARKS

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

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

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

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