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

PowerShell Get-Help on command 'Start-Transcript'

PS C:\Users\wahid> Get-Help Start-Transcript

NAME

Start-Transcript

SYNOPSIS

Creates a record of all or part of a PowerShell session to a text file.

SYNTAX

Start-Transcript [[-LiteralPath] <System.String>] [-Append] [-Force]

[-IncludeInvocationHeader] [-NoClobber] [-Confirm] [-WhatIf]

[<CommonParameters>]

Start-Transcript [[-OutputDirectory] <System.String>] [-Append] [-Force]

[-IncludeInvocationHeader] [-NoClobber] [-Confirm] [-WhatIf]

[<CommonParameters>]

Start-Transcript [[-Path] <System.String>] [-Append] [-Force]

[-IncludeInvocationHeader] [-NoClobber] [-Confirm] [-WhatIf]

[<CommonParameters>]

DESCRIPTION

The `Start-Transcript` cmdlet creates a record of all or part of a PowerShell session to a text file. The transcript includes all command that the user types and all output that appears on the console.

By default, `Start-Transcript` stores the transcript in the following location using the default name:

- Default location: `$HOME\Documents`

- Default filename: `PowerShell_transcript.<computername>.<random>.<timestamp>.txt`

Starting in Windows PowerShell 5.0, `Start-Transcript` includes the hostname in the generated file name of all transcripts. The filename also includes random characters in names to prevent potential overwrites or duplication when you start two or more transcripts simultaneously. Including the computer name is useful if you store your transcripts in a centralized location. The random character string prevents guessing of the filename to gain unauthorized access to the file.
When using the Append parameter, if the target file doesn't have a Byte Order Mark (BOM) `Start-Transcript` defaults to `ASCII` encoding in the target file. This behavior can result in improper encoding of multibyte characters in the transcript.

PARAMETERS

-Append <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet adds the new transcript to the end of an existing file. Use the Path parameter to specify the file.

-Force <System.Management.Automation.SwitchParameter>

Allows the cmdlet to append the transcript to an existing read-only file. When used on a read-only file, the cmdlet changes the file permission to read-write. The cmdlet can't override security restrictions when this parameter is used.

-IncludeInvocationHeader <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet logs the time stamp when commands are run.

-LiteralPath <System.String>

Specifies a location to the transcript file. Unlike the Path parameter, the value of the LiteralPath parameter is used exactly as it's typed. No characters are interpreted as wildcards. If the path includes escape characters, enclose it in single quotation marks. Single quotation marks inform PowerShell not to interpret any characters as escape sequences.

-NoClobber <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet doesn't overwrite an existing file. By default, if a transcript file exists in the specified path, `Start-Transcript` overwrites the file without warning.

-OutputDirectory <System.String>

Specifies a specific path and folder in which to save a transcript. PowerShell automatically assigns the transcript name.

-Path <System.String>

Specifies a location to the transcript file. Enter a path to a `.txt` file. Wildcards aren't permitted. If any of the directories in the path don't exist, the command fails.

If you don't specify a path, `Start-Transcript` uses the path in the value of the `$Transcript` global variable. If you haven't created this variable, `Start-Transcript` stores the transcripts in the default

location and filename.

-Confirm <System.Management.Automation.SwitchParameter>
Prompts you for confirmation before running the cmdlet.

-WhatIf <System.Management.Automation.SwitchParameter>
Shows what would happen if the cmdlet runs. The cmdlet isn't run.

<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: Start a transcript file with default settings ---

Start-Transcript

This command starts a transcript in the default file location.
-- Example 2: Start a transcript file at a specific location --

Start-Transcript -Path "C:\transcripts\transcript0.txt" -NoClobber

This command starts a transcript in the `Transcript0.txt` file in
`C:\transcripts`. NoClobber parameter prevents any existing files from being
overwritten. If the `Transcript0.txt` file already exists, the command fails.
Example 3: Start a transcript file with a unique name and store it on a file
share

```
$sharepath  = '\\Server01\Transcripts'
$username   = $env:USERNAME
$hostname   = hostname
$version    = $PSVersionTable.PSVersion.ToString()
```

```
    $datetime   = Get-Date -f 'yyyyMMddHHmmss'
    $filename   = "Transcript-${username}-${hostname}-${version}-${datetime}.txt"
    $Transcript = Join-Path -Path $sharepath -ChildPath $filename
    Start-Transcript
```

The full path to the transcript file is stored in the `$Transcript` preference variable. For more information about the `$Transcript` preference variable, see about_Preference_Variables (../Microsoft.PowerShell.Core/About/about_Preference_Variables.md#transcript).

REMARKS

To see the examples, type: "get-help Start-Transcript -examples".

For more information, type: "get-help Start-Transcript -detailed".

For technical information, type: "get-help Start-Transcript -full".

For online help, type: "get-help Start-Transcript -online"