



python



PowerShell

FPDF Library
PDF generator

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

PowerShell Get-Help on command 'Add-History'

PS C:\Users\wahid> Get-Help Add-History

NAME

Add-History

SYNOPSIS

Appends entries to the session history.

SYNTAX

```
Add-History [[-InputObject] <System.Management.Automation.PSObject[]>]
[-Passthru] [<CommonParameters>]
```

DESCRIPTION

The `Add-History` cmdlet adds entries to the end of the session history, that is, the list of commands entered during the current session.

The session history is a list of the commands entered during the session. The session history represents the order of execution, the status, and the start and end times of the command. As you enter each command, PowerShell adds it to the history so that you can reuse it. For more information about the session history, see `about_History` (About/about_History.md).

The session history is managed separately from the history maintained by the PSReadLine module. Both histories are available in sessions where PSReadLine is loaded. This cmdlet only works with the session history. For more information see, `about_PSReadLine` (`../PSReadLine/About/about_PSReadLine.md`).

You can use the ``Get-History`` cmdlet to get the commands and pass them to ``Add-History``, or you can export the commands to a CSV or XML file, then import the commands, and pass the imported file to ``Add-History``. You can use this cmdlet to add specific commands to the history or to create a single history file that includes commands from more than one session.

PARAMETERS

`-InputObject <System.Management.Automation.PSObject[]>`

Specifies an array of entries to add to the history as `HistoryInfo` object to the session history. You can use this parameter to submit a `HistoryInfo` object, such as the ones that are returned by the ``Get-History``, ``Import-Clixml``, or ``Import-Csv`` cmdlets, to ``Add-History``.

`-Passthru <System.Management.Automation.SwitchParameter>`

Indicates that this cmdlet returns a `HistoryInfo` object for each history entry. By default, this cmdlet does not generate any output.

<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: Add commands to the history of a different session

```
Import-Csv -Path C:\testing\history.csv | Add-History
```

The first command gets objects representing the commands in the history and exports them to the `History.csv` file.

The second command is typed at the command line of a different session. It uses the `Import-Csv` cmdlet to import the objects in the `History.csv` file.

The pipeline operator (`|`) passes the objects to the `Add-History` cmdlet, which adds the objects representing the commands in the `History.csv` file to the current session history.

----- Example 2: Import and run commands -----

```
Import-Clixml -Path C:\temp\history.xml | Add-History -PassThru |  
ForEach-Object -Process {Invoke-History}
```

The first command uses the `Import-Clixml` cmdlet to import a command history that was exported to the `History.xml` file. The pipeline operator passes the commands to the `Add-History` cmdlet, which adds the commands to the current session history. The `PassThru` parameter passes the objects representing the added commands down the pipeline.

The command then uses the `ForEach-Object` cmdlet to apply the `Invoke-History` command to each of the commands in the combined history. The `Invoke-History` command is formatted as a script block, enclosed in braces (`{}`), as required by the `Process` parameter of the `ForEach-Object` cmdlet.

Example 3: Add commands in the history to the end of the history

```
Get-History -Id 5 -Count 5 | Add-History
```

The `Get-History` cmdlet gets the five commands ending in command 5. The pipeline operator passes them to the `Add-History` cmdlet, which appends them to the current history. The `Add-History` command does not include any parameters, but PowerShell associates the objects passed through the pipeline

with the InputObject parameter of `Add-History`.

Example 4: Add commands in a .csv file to the current history

```
$a = Import-Csv -Path C:\testing\history.csv
```

```
Add-History -InputObject $a -PassThru
```

The `Import-Csv` cmdlet imports the commands in the `History.csv` file and store its contents in the variable `\$a`.

The second command uses the `Add-History` cmdlet to add the commands from `History.csv` to the current session history. It uses the InputObject parameter to specify the `\$a` variable and the PassThru parameter to generate an object to display at the command line. Without the PassThru parameter, the `Add-History` cmdlet does not generate any output.

Example 5: Add commands in an .xml file to the current history

```
Add-History -InputObject (Import-Clixml -Path C:\temp\history.xml)
```

The InputObject parameter passes the results of the command in parentheses to the `Add-History` cmdlet. The command in parentheses, which is executed first, imports the `history.xml` file into PowerShell. The `Add-History` cmdlet then adds the commands in the file to the session history.

REMARKS

To see the examples, type: "get-help Add-History -examples".

For more information, type: "get-help Add-History -detailed".

For technical information, type: "get-help Add-History -full".

For online help, type: "get-help Add-History -online"