Windows PowerShell Get-Help on Cmdlet 'Set-ItemProperty'

PS:\>Get-HELP Set-ItemProperty -Full

NAME

Set-ItemProperty

SYNOPSIS

Creates or changes the value of a property of an item.

SYNTAX

Set-ItemProperty [-Path] <System.String[]> [-Credential <System.Management.Automation.PSCredential>] [-Exclude <System.String[]>] [-Filter <System.String>] [-Force]

[-Include <System.String[]>] -InputObject <System.Management.Automation.PSObject> [-PassThru] [-Type <Microsoft.Win32.RegistryValueKind>] [-UseTransaction] [-Confirm]

[-WhatIf] [<CommonParameters>]

Set-ItemProperty [-Credential <System.Management.Automation.PSCredential>] [-Exclude <System.String[]>] [-Filter <System.String>] [-Force] [-Include

<System.String[]>] -InputObject <System.Management.Automation.PSObject> -LiteralPath <System.String[]> [-PassThru] [-Type <Microsoft.Win32.RegistryValueKind>]

[-UseTransaction] [-Confirm] [-WhatIf] [<CommonParameters>]

Set-ItemProperty [-Name] <System.String> [-Value] <System.Object> [-Credential <System.Management.Automation.PSCredential>] [-Exclude <System.String[]>] [-Filter <System.String>] [-Force] [-Include <System.String[]>] -LiteralPath <System.String[]> [-PassThru] [-Type <Microsoft.Win32.RegistryValueKind>] [-UseTransaction] [-Confirm] [-WhatIf] [<CommonParameters>]

Set-ItemProperty [-Path] <System.String[]> [-Name] <System.String> [-Value] <System.Object> [-Credential <System.Management.Automation.PSCredential>] [-Exclude <System.String[]>] [-Filter <System.String>] [-Force] [-Include <System.String[]>] [-PassThru] [-Type <Microsoft.Win32.RegistryValueKind>] [-UseTransaction] [-Confirm] [-WhatIf] [<CommonParameters>]

DESCRIPTION

The `Set-ItemProperty` cmdlet changes the value of the property of the specified item. You can use the cmdlet to establish or change the properties of items. For example, you can use `Set-ItemProperty` to set the value of the IsReadOnly property of a file object to `$True`.

You also use `Set-ItemProperty` to create and change registry values and data. For example, you can add a new registry entry to a key and establish or change its value.

PARAMETERS

-Credential <System.Management.Automation.PSCredential>

> [!NOTE] > This parameter is not supported by any providers installed with PowerShell. > To impersonate another user, or elevate your credentials when running this cmdlet, > use Invoke-Command (../Microsoft.PowerShell.Core/Invoke-Command.md).

Required? false

Position? named

Default value Current user

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-Exclude <System.String[]>

Specifies, as a string array, an item or items that this cmdlet excludes in the operation. The value of this parameter qualifies the Path parameter. Enter a path

element or pattern, such as ` .txt`. Wildcard characters are permitted. The Exclude * parameter is effective only when the command includes the contents of an

item, such as `C:\Windows*`, where the wildcard character specifies the contents of the `C:\Windows` directory.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? true

-Filter <System.String>

Specifies a filter to qualify the Path parameter. The FileSystem (../Microsoft.PowerShell.Core/About/about_FileSystem_Provider.md)provider is the only installed

PowerShell provider that supports the use of filters. You can find the syntax for the FileSystem filter language in about_Wildcards

(../Microsoft.PowerShell.Core/About/about_Wildcards.md). Filters are more efficient than other parameters, because the provider applies them when the cmdlet gets

the objects rather than having PowerShell filter the objects after they are retrieved.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? true

-Force <System.Management.Automation.SwitchParameter>

Forces the cmdlet to set a property on items that cannot otherwise be accessed by the user. Implementation varies by provider. For more information, see

about_Providers (../Microsoft.PowerShell.Core/About/about_Providers.md).

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-Include <System.String[]>

Specifies, as a string array, an item or items that this cmdlet includes in the operation. The value of this parameter qualifies the Path parameter. Enter a path

element or pattern, such as `" .txt"`. Wildcard characters are permitted. The Include * parameter is effective only when the command includes the contents of an

item, such as `C:\Windows*`, where the wildcard character specifies the contents of the `C:\Windows` directory.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? true

-InputObject <System.Management.Automation.PSObject>

Specifies the object that has the properties that this cmdlet changes. Enter a variable that contains the object or a command that gets the object.

Required? true

Position? named

Default value None

Accept pipeline input? True (ByPropertyName, ByValue)

Accept wildcard characters? false

-LiteralPath <System.String[]>

Specifies a path to one or more locations. The value of LiteralPath is used exactly as it is typed. No characters are

interpreted as wildcards. If the path

includes escape characters, enclose it in single quotation marks. Single quotation marks tell PowerShell not to interpret any characters as escape sequences.

For more information, see about_Quoting_Rules (../Microsoft.Powershell.Core/About/about_Quoting_Rules.md).

Required? true

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-Name <System.String>

Specifies the name of the property.

Required? true

Position? 1

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-PassThru <System.Management.Automation.SwitchParameter>

Returns an object that represents the item property. By default, this cmdlet does not generate any output.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-Path <System.String[]>

Specifies the path of the items with the property to modify. Wildcard characters are permitted.

Required? true

Position? 0

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? true

-Type <Microsoft.Win32.RegistryValueKind>

This is a dynamic parameter made available by the Registry provider. The Registry provider and this parameter are only available on Windows.

Specifies the type of property that this cmdlet adds. The acceptable values for this parameter are:

- `String`: Specifies a null-terminated string. Used for REG_SZ values. - `ExpandString`: Specifies a null-terminated string that contains unexpanded references

to environment variables that are expanded when the value is retrieved. Used for REG_EXPAND_SZ values. - `Binary`: Specifies binary data in any form. Used for

REG_BINARY values. - `DWord`: Specifies a 32-bit binary number. Used for REG_DWORD values. - `MultiString`: Specifies an array of null-terminated strings

terminated by two null characters. Used for REG_MULTI_SZ values. - `Qword`: Specifies a 64-bit binary number. Used for REG_QWORD values. - `Unknown`: Indicates

an unsupported registry data type, such as REG_RESOURCE_LIST values.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-UseTransaction [<SwitchParameter>]

Includes the command in the active transaction. This parameter is valid only when a transaction is in progress. For more information, see about_Transactions

(../Microsoft.PowerShell.Core/About/about_Transactions.md).

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-Value <System.Object>

Specifies the value of the property.

Required? true

Position? 2

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-Confirm <System.Management.Automation.SwitchParameter>

Prompts you for confirmation before running the cmdlet.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-WhatIf <System.Management.Automation.SwitchParameter>

Shows what would happen if the cmdlet runs. The cmdlet is not run.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

<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).

INPUTS

System.Management.Automation.PSObject

You can pipe objects to this cmdlet.

OUTPUTS

None

By default, this cmdlet returns no output.

System.Management.Automation.PSCustomObject

When you use the PassThru parameter, this cmdlet returns a PSCustomObject object representing the item that was changed and its new property value.

NOTES

Windows PowerShell includes the following aliases for `Set-ItemProperty`:

- `sp`

`Set-ItemProperty` is designed to work with the data exposed by any provider. To list the providers available in your session, type `Get-PSProvider`. For more

information, see about_Providers (../Microsoft.PowerShell.Core/About/about_Providers.md).

---------- Example 1: Set a property of a file ----------

```
Set-ItemProperty -Path C:\GroupFiles\final.doc -Name IsReadOnly -Value $true
```

--------- Example 2: Create a registry entry and value ---------

```
Set-ItemProperty -Path "HKLM:\Software\ContosoCompany" -Name "NoOfEmployees" -Value 823
Get-ItemProperty -Path "HKLM:\Software\ContosoCompany"

PSPath        : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\software\contosocompany
PSParentPath  : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\software
PSChildName   : contosocompany
PSDrive       : HKLM
PSProvider    : Microsoft.PowerShell.Core\Registry
NoOfLocations : 2
NoOfEmployees : 823

Set-ItemProperty -Path "HKLM:\Software\ContosoCompany" -Name "NoOfEmployees" -Value 824
Get-ItemProperty -Path "HKLM:\Software\ContosoCompany"

PSPath        : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\software\contosocompany
PSParentPath  : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\software
PSChildName   : contosocompany
PSDrive       : HKLM
PSProvider    : Microsoft.PowerShell.Core\Registry
NoOfLocations : 2
NoOfEmployees : 824
```

The first command creates the registry entry. It uses Path to specify the path of the `HKLM:` drive and the `Software\MyCompany` key. The command uses Name to specify
the entry name and Value to specify a value.

The second command uses the `Get-ItemProperty` cmdlet to see the new registry entry. If you use the `Get-Item` or `Get-ChildItem` cmdlets, the entries do not appear

because they are properties of a key, not items or child items.

The third command changes the value of the NoOfEmployees entry to 824.

You can also use the `New-ItemProperty` cmdlet to create the registry entry and its value and then use `Set-ItemProperty` to change the value.

For more information about the `HKLM:` drive, type `Get-Help Get-PSDrive`. For more information about how to use PowerShell to manage the registry, type `Get-Help

Registry`.

------- Example 3: Modify an item by using the pipeline -------

Get-ChildItem weekly.txt | Set-ItemProperty -Name IsReadOnly -Value $True

RELATED LINKS

Online Version: https://learn.microsoft.com/powershell/module/microsoft.powershell.management/set-itemproperty?view=powershell-5.1&WT.mc_id=ps-gethelp

Clear-ItemProperty

Copy-ItemProperty

Get-ItemProperty

Move-ItemProperty

New-ItemProperty

Remove-ItemProperty

Rename-ItemProperty

about_Providers