



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 'Set-ItemProperty'**

**PS C:\Users\wahid> Get-Help Set-ItemProperty**

#### 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`).

`-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.

**-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.

**-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)`.

**-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.

`-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.

`-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`).

`-Name <System.String>`

Specifies the name of the property.

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

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

`-Path <System.String[]>`

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

`-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.

#### -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).

#### -Value <System.Object>

Specifies the value of the property.

#### -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 is not 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>).

```
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
```

## REMARKS

To see the examples, type: "get-help Set-ItemProperty -examples".

For more information, type: "get-help Set-ItemProperty -detailed".

For technical information, type: "get-help Set-ItemProperty -full".

For online help, type: "get-help Set-ItemProperty -online"