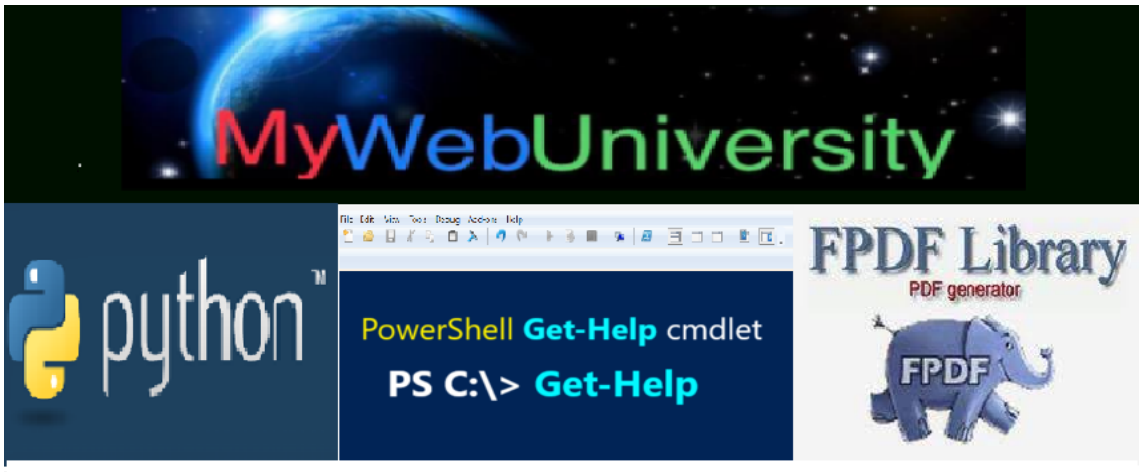

Full credit is given to all the above companies including the Operating System that this PDF file was generated!

Windows PowerShell Get-Help on Cmdlet 'New-Item'

PS:\>Get-HELP New-Item -Full

NAME

New-Item

SYNOPSIS

Creates a new item.

SYNTAX

New-Item [[-Path] <System.String[]>] [-ApplicationName <System.String>] [-Authentication <Microsoft.WSMan.Management.AuthenticationMechanism>] [-CertificateThumbprint <System.String>] [-Credential <System.Management.Automation.PSCredential>] [-Force] [-ItemType <System.String>] -Name <System.String> [-Options <System.Management.Automation.ScopedItemOptions>] [-OptionSet <System.Collections.Hashtable>] [-Port <System.Int32>] [-SessionOption <Microsoft.WSMan.Management.SessionOption>] [-UseSSL] [-UseTransaction] [-Value <System.Object>] [-Confirm] [-WhatIf] [<CommonParameters>]

New-Item [[-Path] <System.String[]>] [-Authentication <Microsoft.WSMan.Management.AuthenticationMechanism>] [-CertificateThumbprint <System.String>] -ConnectionURI <System.Uri> [-Credential <System.Management.Automation.PSCredential>] [-Force] [-ItemType <System.String>]

[-Options

<System.Management.Automation.ScopedItemOptions>] [-OptionSet <System.Collections.Hashtable>] [-Port <System.Int32>] [-SessionOption

<Microsoft.WSMan.Management.SessionOption>] [-UseTransaction] [-Value <System.Object>] [-Confirm] [-WhatIf] [<CommonParameters>]

DESCRIPTION

The `New-Item` cmdlet creates a new item and sets its value. The types of items that can be created depend on the location of the item. For example, in the file

system, `New-Item` creates files and folders. In the registry, `New-Item` creates registry keys and entries.

`New-Item` can also set the value of the items that it creates. For example, when it creates a new file, `New-Item` can add initial content to the file.

PARAMETERS

-ApplicationName <System.String>

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

Specifies the application name in the connection. The default value of the ApplicationName parameter is WSMAN .

For more information, see New-WSManInstance (../Microsoft.WSMan.Management/New-WSManInstance.md).

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Authentication <Microsoft.WSMan.Management.AuthenticationMechanism>

This is a dynamic parameter made available by the WSMan provider. The WSMan provider and this parameter are

only available on Windows.

Specifies the authentication mechanism to be used at the server.

For more information, see New-WSManInstance (../Microsoft.WSMan.Management/New-WSManInstance.md).

Required? false
Position? named
Default value None
Accept pipeline input? False
Accept wildcard characters? false

-CertificateThumbprint <System.String>

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

Specifies the digital public key certificate (X509) of a user account that has permission to perform this WSMan action. Enter the certificate thumbprint of the
certificate.

For more information, see New-WSManInstance (../Microsoft.WSMan.Management/New-WSManInstance.md).

Required? false
Position? named
Default value None
Accept pipeline input? False
Accept wildcard characters? false

-ConnectionURI <System.Uri>

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

Specifies the connection endpoint for WSMan.

For more information, see New-WSManInstance (../Microsoft.WSMan.Management/New-WSManInstance.md).

Required? true

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Credential <System.Management.Automation.PSCredential>

> [!NOTE] > This parameter isn't supported by any providers installed with PowerShell. To impersonate another > user or elevate your credentials when running this

cmdlet, use `Invoke-Command`.

Required? false

Position? named

Default value Current user

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-Force <System.Management.Automation.SwitchParameter>

Forces this cmdlet to create an item that writes over an existing read-only item. Implementation varies from provider to provider. Even using the Force parameter,

the cmdlet can't override security restrictions.

You can't use Force to overwrite an existing Junction. Attempts to overwrite an existing Junction fail with a "cannot be removed because it is not empty" error.

You must remove the existing Junction before creating a new one.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-ItemType <System.String>

Specifies the provider-specified type of the new item. The available values of this parameter depend on the current provider you are using.

If your location is in a `FileSystem` drive, the following values are allowed:

- File

- Directory

- SymbolicLink

- Junction

- HardLink

When you create a file using this method, the resulting file is encoded as UTF-8 without a byte-order-mark (BOM).

In a `Certificate` drive, these are the values you can specify:

- Certificate Provider

- Certificate

- Store

- StoreLocation

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

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-Name <System.String>

Specifies the name of the new item. You can specify the name of the new item in the Name or Path parameter value, and you can specify the path of the new item in

Name or Path value. Items names passed using the Name parameter are created relative to the value of the Path parameter.

Required? true

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-Options <System.Management.Automation.ScopedItemOptions>

This is a dynamic parameter made available by the Alias provider. For more information, see New-Alias (../Microsoft.PowerShell.Utility/New-Alias.md).

Specifies the value of the Options property of an alias.

Valid values are:

- `None`: The alias has no constraints (default value)

- `ReadOnly`: The alias can be deleted but can't be changed without using the Force parameter - `Constant`: The alias can't be deleted or changed

- `Private`: The alias is available only in the current scope

- `AllScope`: The alias is copied to any new scopes that are created

- `Unspecified`: The option isn't specified

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-OptionSet <System.Collections.Hashtable>

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

Passes a set of switches to a service to modify or refine the nature of the request.

For more information, see New-WSManInstance (../Microsoft.WSMan.Management/New-WSManInstance.md).

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Path <System.String[]>

Specifies the path of the location of the new item. The default is the current location when Path is omitted. You can specify the name of the new item in Name ,

or include it in Path . Items names passed using the Name parameter are created relative to the value of the Path parameter.

For this cmdlet, the Path parameter works like the LiteralPath parameter of other cmdlets. Wildcard characters are not interpreted. All characters are passed to

the location's provider. The provider may not support all characters. For example, you can't create a filename that contains an asterisk (`*`) character.

Required? false

Position? 0

Default value Current location

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-Port <System.Int32>

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

Specifies the port to use when the client connects to the WinRM service.

For more information, see New-WSManInstance (../Microsoft.WSMan.Management/New-WSManInstance.md).

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-SessionOption <Microsoft.WSMan.Management.SessionOption>

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

Defines a set of extended options for the WS-Management session.

For more information, see New-WSManInstance (../Microsoft.WSMan.Management/New-WSManInstance.md).

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-UseSSL <System.Management.Automation.SwitchParameter>

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

Specifies that the Secure Sockets Layer (SSL) protocol should be used to establish a connection to the remote computer. By default, SSL isn't used.

For more information, see New-WSManInstance (../Microsoft.WSMan.Management/New-WSManInstance.md).

Required? false

Position? named

Default value False

Accept pipeline input? ByValue (False), ByName (False)

Accept wildcard characters? false

-UseTransaction <System.Management.Automation.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 new item. You can also pipe a value to `New-Item`.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName, ByValue)

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 isn't 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.Object

You can pipe a value for the new item to this cmdlet.

OUTPUTS

System.Collections.DictionaryEntry

The cmdlet returns a DictionaryEntry object when creating a new environment variable.

System.IO.DirectoryInfo

The cmdlet returns a DirectoryInfo object when creating a new directory in the filesystem.

System.IO.FileInfo

The cmdlet returns a FileInfo object when creating a new file in the filesystem.

System.Management.Automation.AliasInfo

The cmdlet returns an AliasInfo object when creating a new alias.

System.Management.Automation.FunctionInfo

The cmdlet returns a FunctionInfo object when creating a new function.

System.Management.Automation.PSVariable

The cmdlet returns a PSVariable object when creating a new variable.

NOTES

Windows PowerShell includes the following aliases for `New-Item`:

- `ni`

`New-Item` 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: Create a file in the current directory ------

New-Item -Path . -Name "testfile1.txt" -ItemType "file" -Value "This is a text string."

---------- Example 2: Create a directory ----------

New-Item -Path "c:\" -Name "logfiles" -ItemType "directory"

---------- Example 3: Create a profile ----------

New-Item -Path $profile -ItemType "file" -Force

> [!NOTE] > When you create a file using this method, the resulting file is encoded as UTF-8 without a > byte-order-mark (BOM).

---- Example 4: Create a directory in a different directory ----

New-Item -ItemType "directory" -Path "c:\ps-test\scripts"

---------- Example 5: Create multiple files ----------

New-Item -ItemType "file" -Path "c:\ps-test\test.txt", "c:\ps-test\Logs\test.log"

Example 6: Use wildcards to create files in multiple directories

Get-ChildItem -Path C:\Temp\

Directory: C:\Temp

Mode LastWriteTime Length Name

```
----              -------------     ------ ----
d-----        5/15/2019   6:45 AM        1   One
d-----        5/15/2019   6:45 AM        1   Two
d-----        5/15/2019   6:45 AM        1   Three

New-Item -Path C:\Temp\* -Name temp.txt -ItemType File | Select-Object FullName

FullName
--------
C:\Temp\One\temp.txt
C:\Temp\Three\temp.txt
C:\Temp\Two\temp.txt
```

The `Get-ChildItem` cmdlet shows three directories under the `C:\Temp` directory. Using wildcards the `New-Item` cmdlet creates a `temp.txt` file in all of the

directories under the current directory. The `New-Item` cmdlet outputs the items you created, which is piped to `Select-Object` to verify the paths of the newly

created files.

---- Example 7: Create a symbolic link to a file or folder ----

```
$link = New-Item -ItemType SymbolicLink -Path .\link -Target .\Notice.txt
$link | Select-Object LinkType, Target

LinkType     Target
--------     ------
SymbolicLink {.\Notice.txt}
```

In this example, Target is an alias for the Value parameter. The target of the symbolic link must be a fully-qualified path.

Example 8: Use the -Force parameter to attempt to recreate folders

```
PS> New-Item -Path .\TestFolder -ItemType Directory
PS> New-Item -Path .\TestFolder\TestFile.txt -ItemType File
```

```
PS> New-Item -Path .\TestFolder -ItemType Directory -Force


    Directory: C:\
Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         5/1/2020   8:03 AM                TestFolder

PS> Get-ChildItem .\TestFolder\


    Directory: C:\TestFolder
Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         5/1/2020   8:03 AM              0 TestFile.txt
```

Example 9: Use the -Force parameter to overwrite existing files

```
PS> New-Item ./TestFile.txt -ItemType File -Value 'This is just a test file'


    Directory: C:\Source\Test
Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         5/1/2020   8:32 AM             24 TestFile.txt

New-Item ./TestFile.txt -ItemType File -Force


    Directory: C:\Source\Test
Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         5/1/2020   8:32 AM              0 TestFile.txt
```

> [!NOTE] > When using `New-Item` with the `-Force` switch to create registry keys, the command will behave > the same as when overwriting a file. If the registry key

already exists, the key and all > properties and values will be overwritten with an empty registry key.

RELATED LINKS

Online Version:

https://learn.microsoft.com/powershell/module/microsoft.powershell.management/new-item?view=powershell-5.1&WT.mc_id=ps-gethelp

Clear-Item

Copy-Item

Get-Item

Invoke-Item

Move-Item

Remove-Item

Rename-Item

Set-Item

about_Providers