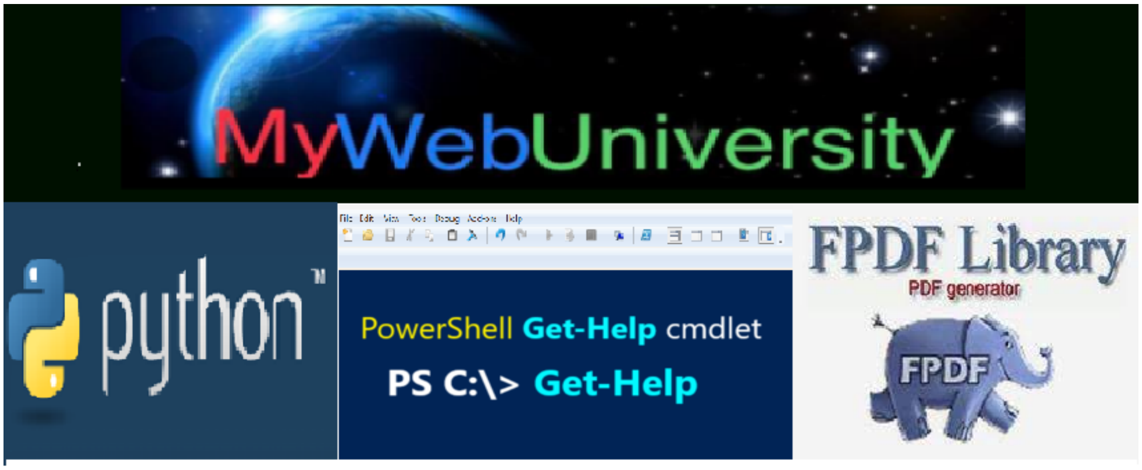

Full credit is given to all the above companies including the Operating System that this PDF file was generated!

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

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

NAME

    Get-Item

SYNOPSIS

    Gets the item at the specified location.

SYNTAX

```
            Get-Item [-CodeSigningCert] [-Credential <System.Management.Automation.PSCredential>] [-DnsName
<Microsoft.PowerShell.Commands.DnsNameRepresentation>]
    [-DocumentEncryptionCert] [-Eku <System.String>] [-Exclude <System.String[]>] [-ExpiringInDays <System.Int32>]
[-Filter <System.String>] [-Force] [-Include
        <System.String[]> -LiteralPath <System.String[]> [-SSLServerAuthentication] [-Stream <System.String[]>]
[-UseTransaction] [<CommonParameters>]

    Get-Item [-Path] <System.String[]> [-CodeSigningCert] [-Credential <System.Management.Automation.PSCredential>]
[-DnsName
      <Microsoft.PowerShell.Commands.DnsNameRepresentation>] [-DocumentEncryptionCert] [-Eku <System.String>]
[-Exclude <System.String[]>] [-ExpiringInDays <System.Int32>]
    [-Filter <System.String>] [-Force] [-Include <System.String[]>] [-SSLServerAuthentication] [-Stream <System.String[]>]
```

[-UseTransaction] [<CommonParameters>]

DESCRIPTION

The `Get-Item` cmdlet gets the item at the specified location. It doesn't get the contents of the item at the location unless you use a wildcard character (`*`) to
request all the contents of the item.

This cmdlet is used by PowerShell providers to navigate through different types of data stores. Some parameters are only available for a specific provider. For more
information, see about_Providers (../Microsoft.PowerShell.Core/About/about_Providers.md).

PARAMETERS

-CodeSigningCert <System.Management.Automation.SwitchParameter>

This is a dynamic parameter made available by the Certificate provider.

To get certificates that have `Code Signing` in their EnhancedKeyUsageList property value, use the CodeSigningCert parameter.

For more information, see about_Certificate_Provider (../Microsoft.PowerShell.Security/About/about_Certificate_Provider.md).

```
Required?                    false
Position?                    named
Default value                False
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 (../Microsoft.PowerShell.Core/Invoke-Command.md).

Required?   false

Position?   named

Default value   Current user

Accept pipeline input?   True (ByPropertyName)

Accept wildcard characters? false

-DnsName <Microsoft.PowerShell.Commands.DnsNameRepresentation>

This is a dynamic parameter made available by the Certificate provider.

Specifies a domain name or name pattern to match with the DNSNameList property of certificates the cmdlet gets. The value of this parameter can either be

`Unicode` or `ASCII`. Punycode values are converted to Unicode. Wildcard characters (`*`) are permitted.

This parameter was reintroduced in PowerShell 7.1

For more information, see about_Certificate_Provider (../Microsoft.PowerShell.Security/About/about_Certificate_Provider.md).

Required?   false

Position?   named

Default value   None

Accept pipeline input?   False

Accept wildcard characters? true

-DocumentEncryptionCert <System.Management.Automation.SwitchParameter>

This is a dynamic parameter made available by the Certificate provider.

To get certificates that have `Document Encryption` in their EnhancedKeyUsageList property value, use the DocumentEncryptionCert parameter.

For more information, see about_Certificate_Provider (../Microsoft.PowerShell.Security/About/about_Certificate_Provider.md).

Required?   false

Position?   named

Default value   False

Accept pipeline input?   False

Accept wildcard characters? false

-Eku <System.String>

This is a dynamic parameter made available by the Certificate provider.

Specifies text or a text pattern to match with the EnhancedKeyUsageList property of certificates the cmdlet gets. Wildcard characters (` `) are permitted. The

EnhancedKeyUsageList * property contains the friendly name and the OID fields of the EKU.

This parameter was reintroduced in PowerShell 7.1

For more information, see about_Certificate_Provider (../Microsoft.PowerShell.Security/About/about_Certificate_Provider.md).

Required?   false

Position?   named

Default value   None

Accept pipeline input?   False

Accept wildcard characters? true

-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

-ExpiringInDays <System.Int32>

This is a dynamic parameter made available by the Certificate provider.

Specifies that the cmdlet should only return certificates that are expiring in or before the specified number of days. A value of zero (`0`) gets certificates

that have expired.

This parameter was reintroduced in PowerShell 7.1

For more information, see about_Certificate_Provider (../Microsoft.PowerShell.Security/About/about_Certificate_Provider.md).

Required?   false

Position?   named

Default value   None

Accept pipeline input?   False

Accept wildcard characters? false

-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 filters. Filters are more efficient than other parameters. The provider applies filter when the cmdlet gets the objects rather

than having PowerShell filter the objects after they're retrieved. The filter string is passed to the .NET API to enumerate files. The API only supports `*` and

`?` wildcards.

Required?   false

Position?   named

Default value   None

Accept pipeline input?   False

Accept wildcard characters? true

-Force <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet gets items that can't otherwise be accessed, such as hidden items. Implementation varies from provider to provider. For more

information, see about_Providers (../Microsoft.PowerShell.Core/About/about_Providers.md). Even using the Force parameter, the cmdlet can't override security

restrictions.

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

-LiteralPath <System.String[]>

Specifies a path to one or more locations. The value of LiteralPath 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 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

-Path <System.String[]>

Specifies the path to an item. This cmdlet gets the item at the specified location. Wildcard characters are permitted. This parameter is required, but the

parameter name Path is optional.

Use a dot (`.`) to specify the current location. Use the wildcard character (`*`) to specify all the items in the current location.

Required?   true

Position?   0

Default value   None

Accept pipeline input?   True (ByPropertyName, ByValue)

Accept wildcard characters? true

-SSLServerAuthentication <System.Management.Automation.SwitchParameter>

This is a dynamic parameter made available by the Certificate provider.

To get certificates that have `Server Authentication` in their EnhancedKeyUsageList property value, use the SSLServerAuthentication parameter.

For more information, see about_Certificate_Provider

(../Microsoft.PowerShell.Security/About/about_Certificate_Provider.md).

```
        Required?                    false
        Position?                    named
        Default value                False
        Accept pipeline input?       False
        Accept wildcard characters?  false
```

-Stream <System.String[]>

This is a dynamic parameter made available by the FileSystem provider.

Gets the specified alternate NTFS file stream from the file. Enter the stream name. Wildcards are supported. To get all streams, use an asterisk (`*`). This

parameter isn't valid on folders.

This parameter was introduced in PowerShell 3.0.

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

```
        Required?                    false
        Position?                    named
        Default value                No alternate file streams
        Accept pipeline input?       False
        Accept wildcard characters?  true
```

-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

<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.String

You can pipe a string that contains a path to this cmdlet.

OUTPUTS

System.Management.Automation.AliasInfo

The cmdlet outputs this type when accessing the `Alias:` drive.

Microsoft.PowerShell.Commands.X509StoreLocation

System.Security.Cryptography.X509Certificates.X509Store

System.Security.Cryptography.X509Certificates.X509Certificate2

The cmdlet outputs these types when accessing the `Cert:` drive.

System.Collections.DictionaryEntry

The cmdlet outputs this type when accessing the `Env:` drive.

System.IO.DirectoryInfo

System.IO.FileInfo

The cmdlet outputs these types when accessing the Filesystem drives.

System.Management.Automation.FunctionInfo

System.Management.Automation.FilterInfo

The cmdlet outputs these types when accessing the `Function:` drives.

Microsoft.Win32.RegistryKey

The cmdlet outputs this type when accessing the Registry drives.

System.Management.Automation.PSVariable

The cmdlet outputs this type when accessing the `Variable:` drives.

Microsoft.WSMan.Management.WSManConfigContainerElement

Microsoft.WSMan.Management.WSManConfigLeafElement

The cmdlet outputs these types when accessing the `WSMan:` drives.

NOTES

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

- `gi`

This cmdlet does not have a Recurse parameter, because it gets only an item, not its contents. To get the contents of an item recursively, use `Get-ChildItem`.

To navigate through the registry, use this cmdlet to get registry keys and the `Get-ItemProperty` to get registry values and data. The registry values are

considered to be properties of the registry key.

This cmdlet 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: Get the current directory -------------

```
Get-Item .

Directory: C:\

Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----        7/26/2006  10:01 AM            ps-test
```

---- Example 2: Get all the items in the current directory ----

```
Get-Item *

Directory: C:\ps-test

Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----        7/26/2006   9:29 AM            Logs
d----        7/26/2006   9:26 AM            Recs
-a---        7/26/2006   9:28 AM         80 date.csv
-a---        7/26/2006  10:01 AM         30 filenoext
-a---        7/26/2006   9:30 AM      11472 process.doc
-a---        7/14/2006  10:47 AM         30 test.txt
```

------- Example 3: Get the current directory of a drive -------

```
Get-Item C:
```

--------- Example 4: Get items in the specified drive ---------

```
Get-Item C:\*
```

In PowerShell, use a single asterisk (` `) to get contents, instead of the traditional ` .*`. The format is interpreted literally, so ` . ` wouldn't retrieve directories or filenames without a dot.

----- Example 5: Get a property in the specified directory -----

```
(Get-Item C:\Windows).LastAccessTime
```

-------- Example 6: Show the contents of a registry key --------

```
Get-Item HKLM:\Software\Microsoft\Powershell\1\Shellids\Microsoft.Powershell\
```

-- Example 7: Get items in a directory that have an exclusion --

```
Get-Item C:\Windows\*.* -Exclude "w*"
```

RELATED LINKS

Online Version: https://learn.microsoft.com/powershell/module/microsoft.powershell.management/get-item?view=powershell-5.1&WT.mc_id

=ps-gethelp

Clear-Item

Copy-Item

Invoke-Item

Move-Item

New-Item

Remove-Item

Rename-Item

Set-Item

Get-ChildItem

Get-ItemProperty

Get-PSProvider

about_Providers